User Guide

Embedded application

Ease_EG-IoT-V2.20

Reference : Ease_EG-IoT_1149_UG_V2.20_UK_0011

Document History

Rev.	Modifications	Author	Date	Validation	Date
000	Creation for V2.20 (Based on 2.15.2)	PBA	21/02/2022	MAI	
001	Update of user guide according to mantis 4854, 5017, 5104, 5221, 5283. Update after V000 remarks	PBA	17/05/2022	MAI	2022-05-18

TABLE OF CONTENTS

Warning

- This document contains the commissioning information samples **EG-IoT xxxx** based on the embedded application **EaseEG-IoT**.
- **Ercogener** cannot be held responsible for:
  - Problems arising from improper use of the **EG-IoT xxxx**.
  - Problems arising from improper configuration.
  - Malfunction arising from the absence or poor coverage of GSM, GPRS, UMTS, LTE Cat.M1, GNSS, LoRa, Sigfox networks
  - Malfunction if the product is used for the monitoring of physical persons where human life is at stake.
- **Ercogener** reserves the right to modify the functionalities without prior notice.
- This document is applicable to the V2.20 revision.

⚠ A 512Ko flash EG-IoT (from EG-IoT xxx-0600) is required for V2.20 version. Please contact **ercogener** if you have any doubt.

Symbols used

The following symbols are used to highlight important information in the manual.

Essential information for the integration and performance of the module.

⚠ Warning indicating actions that could harm or damage the module.
Warning indicating actions that have an expected impact on the functionalities, autonomous or additional cost with operators (LPWAN, cellular).

Copyright

The use of certain products or services described in this document may require subscription to a paid service. The availability of certain products or services described in this document may vary depending on the configuration and hardware.
In some countries, there may be restrictions on the use of devices. Check with the local legal authorities.

The contents of this document are provided "as is". Except as required by applicable laws in Europe, no warranty of any kind, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, is granted as to the accuracy, reliability or content of the document. **Ercogener** reserves the right to revise this document or withdraw it at any time without notice.

Ercogener cannot be held responsible for any loss of data or income, as well as for any special, incidental, consequential or indirect damage.

1 Principles of Ease_EG-IoT

Two distinct operating modes are configurable: tracking mode and remote reading mode.

Features:

- Management of date and time
- Management of Commissioning (startup): Immediate, Delay, Placed on the clip
- Management of keep-alive frame
- Management of transmission mode: LoRa, Sigfox, Cellular (2G / 3G / LTE-M / NB-IoT) for transmission of SMS and/or TCP/UDP/MQTT/MQTTS frame
- Management of internal sensors (accelerometer, magnetometer, temperature)
- Management of 1 digital output
- Management of BLE advertising mode.
- Management of shock detection
- Management of angular deviation detection
- and

- For tracking
  - Management of position report
  - Management of geofence alerts
  - Management of movement (start and stop)
  - Management of cyclic transmission during movement
  - Read the 2 digital inputs
  - Read the 2 analog inputs (Cellular only)

- For remote reading
  - Management of the 2 digital inputs (pulse counting, horometer or edge detection)
  - Management of thresholds for counting (high level)
  - Management of thresholds for temperature at each transmission (low level and high level)
  - Management of digital output (with downlink)

The fixed limits not indicated in the menu are:

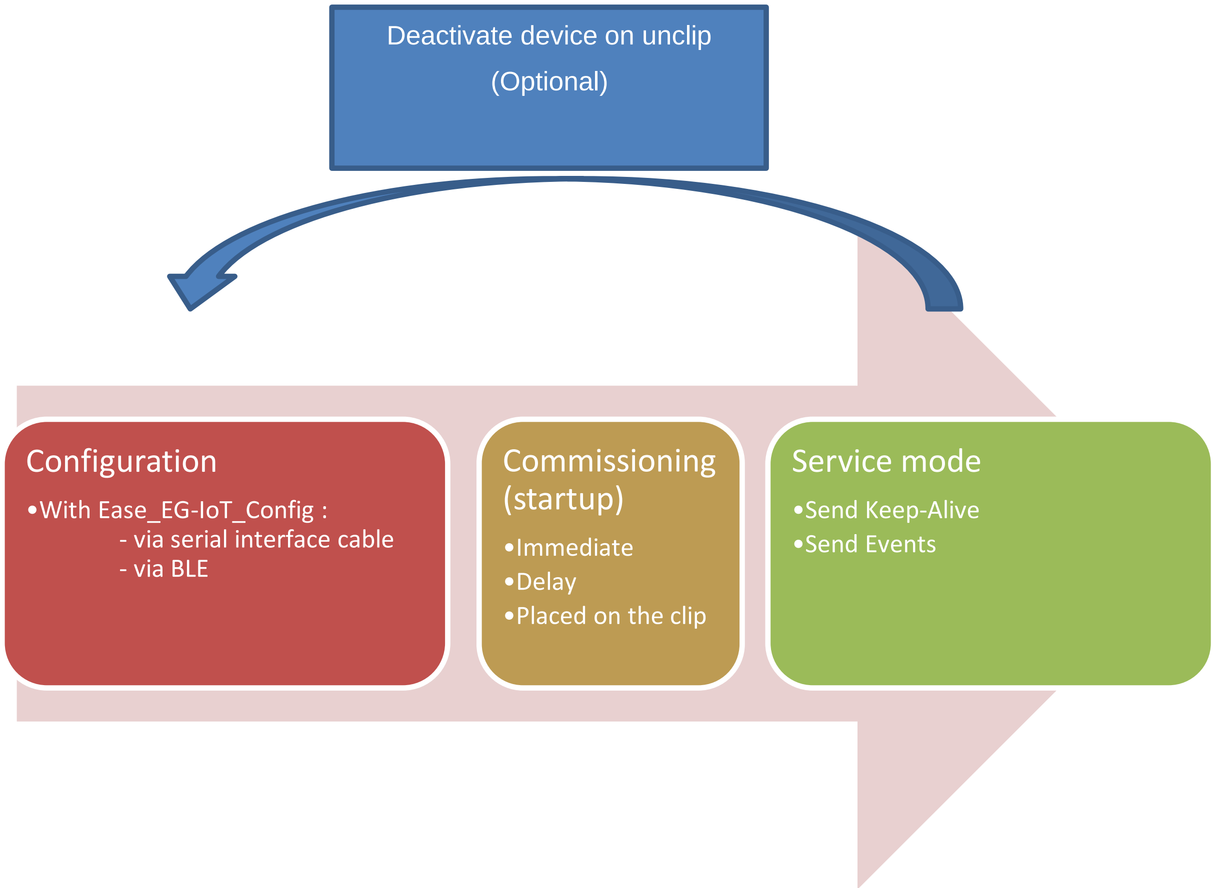
- Time : UTC time
- Temperature resolution : 1° C
- Angle resolution ± 2°
- Hysteresis of geofence : ± 50m
- Hysteresis of temperature: 0 / +2°C
- Resolution of horometer: 1 sec, (sending values in payload in minute)
- Debouncing filter for inputs (edge detection, pulse counter, horometer) 10 ms to 10 sec : default 10 ms

To reduce power consumption, the product wakes up:

- At the Keep-alive frequency and at the alert cyclic frequency.
- Upon an asynchronous event: movement start, shock or edge detection.

The device menu access can be protected by a login and a password.

1.1 Operating cycle



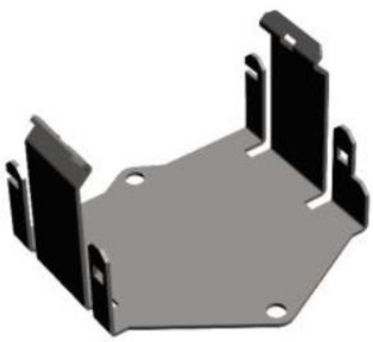


Figure 1: EG-IoT Clip

2 Application in tracking mode:

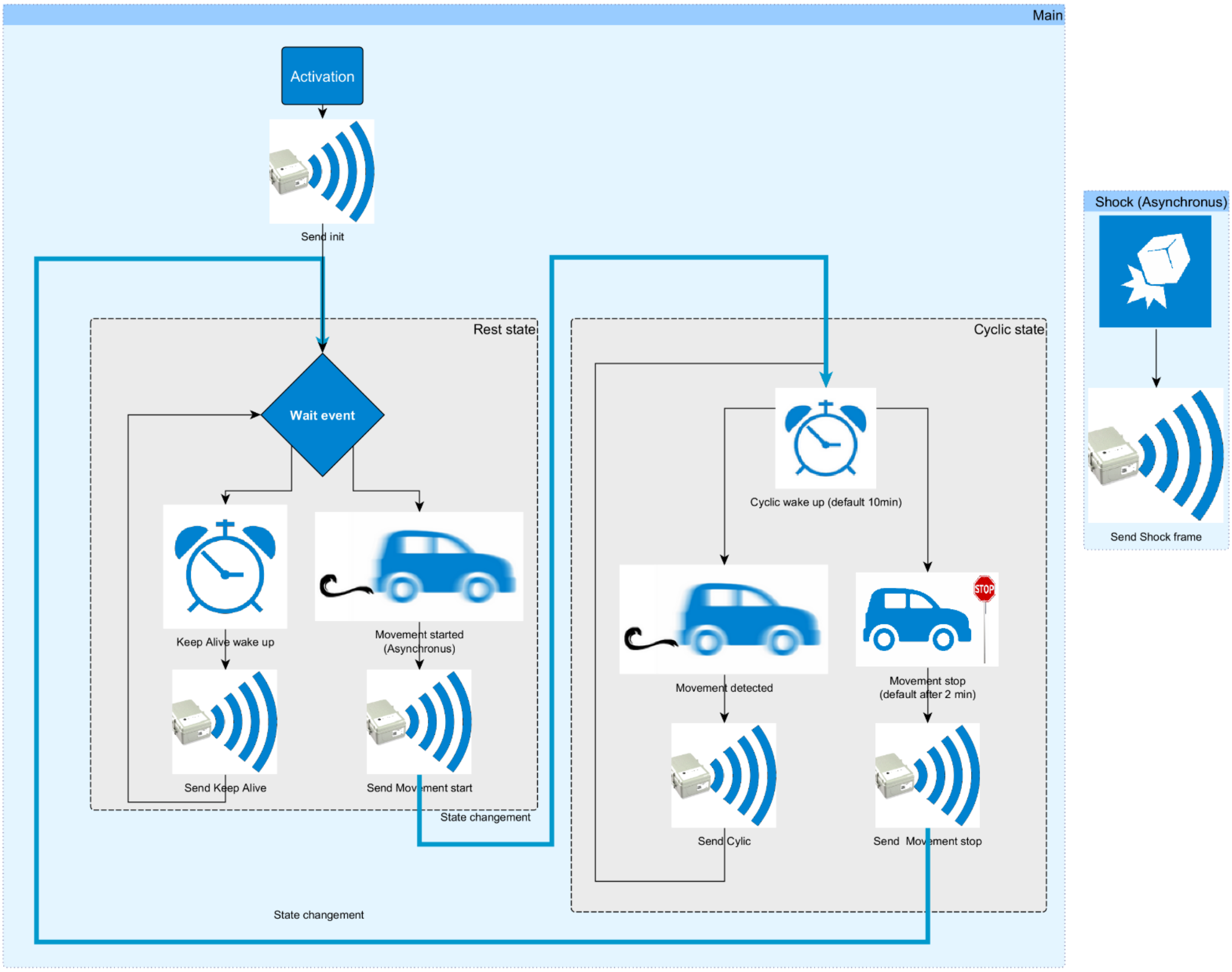
2.1 Operating timing and frame types

In tracking mode, the device sends its localisation and other information regularly.
We can distinguish 2 different functioning modes, rest state and cyclic state.

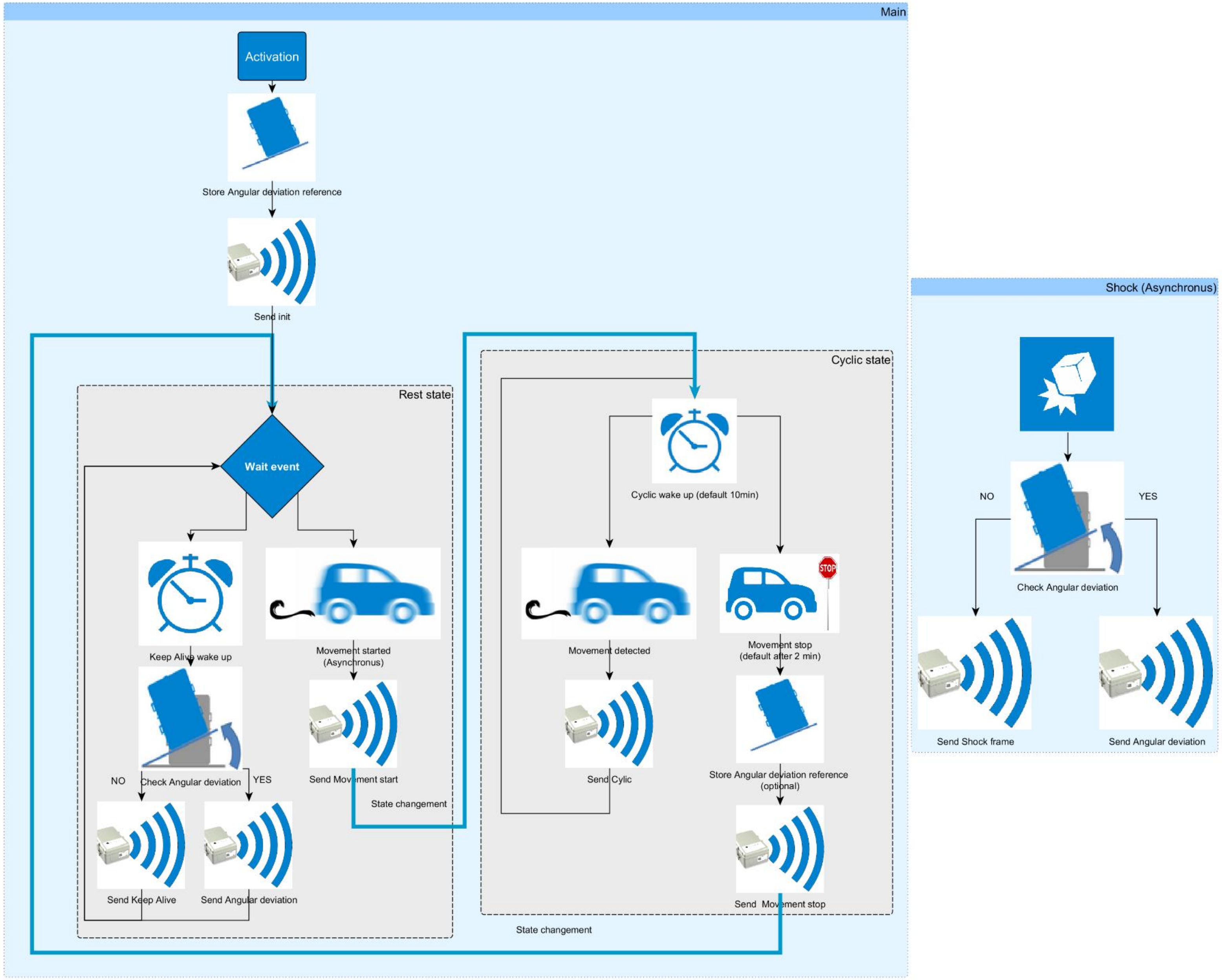
In rest state, the device sends frame at the keep alive frequency.
In movement, the device sends frame at the cyclic frequency (if enabled)

The shock alarm is independent of the state.

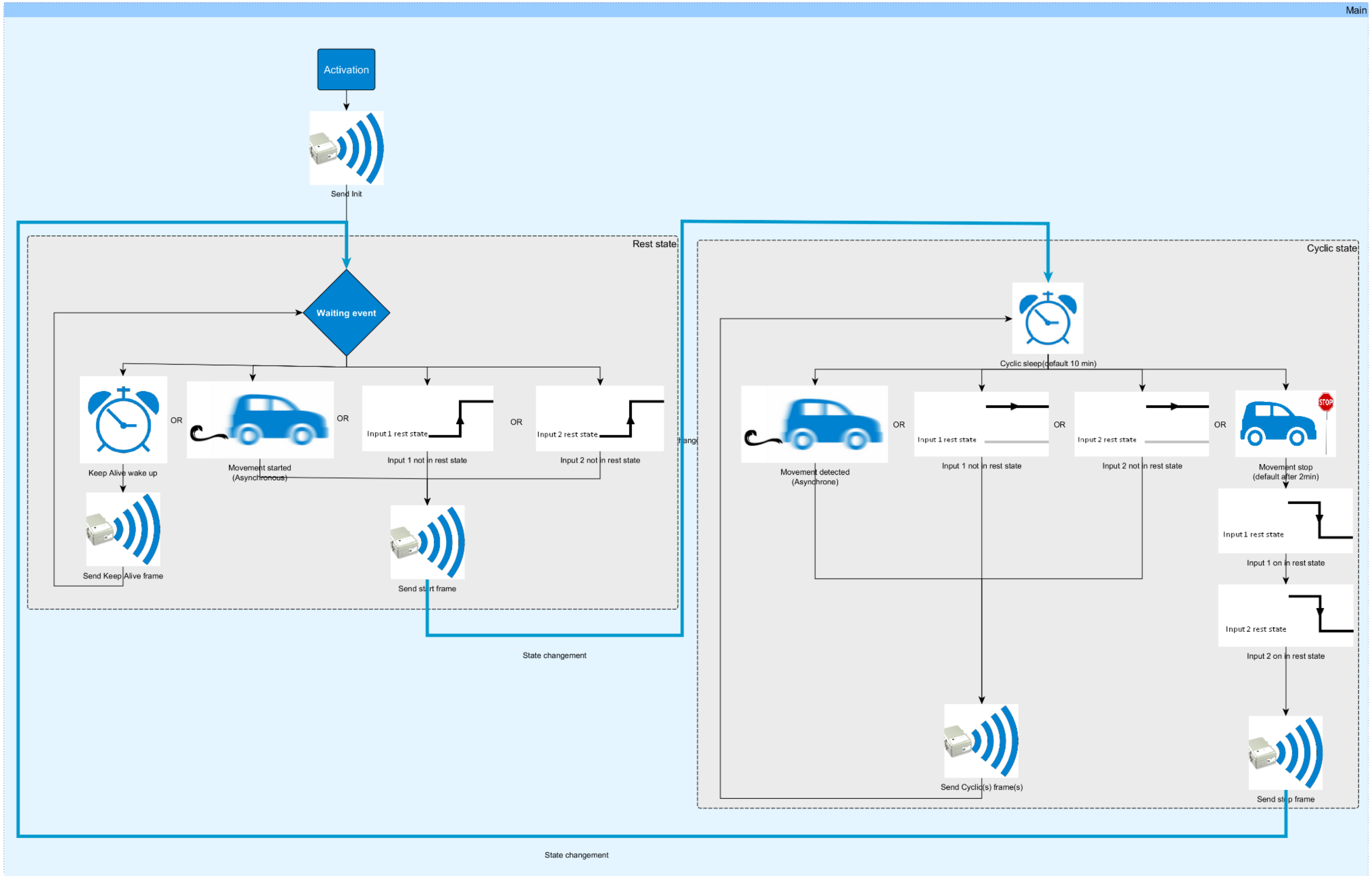
Movement monitoring only

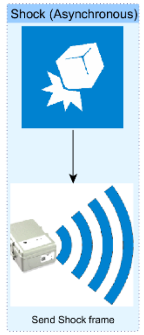


Movement monitoring with angular deviation detection

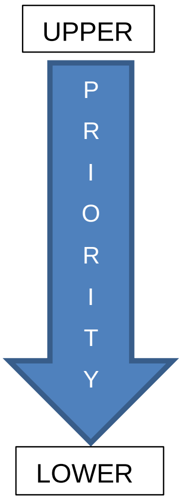


Movement and inputs monitoring



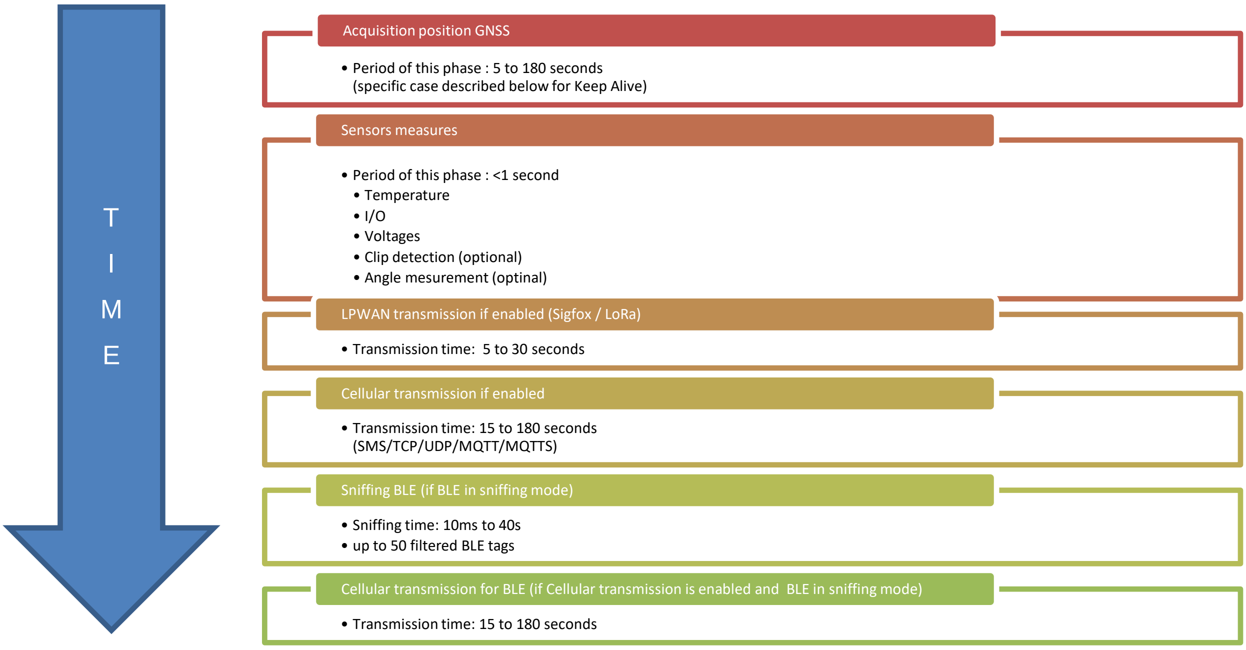


The application can send several types of frame. Each frame has a field called Opcode and its value depends on the functional state.



Opcode	Label	Details
0x00	Initialisation	Indicates that the device was activated or restarted
0xFF	Exit service mode	Stop application
0x02	LPWAN reset	Reset due to LPWAN downlink
0X86	Angular deviation	Inclination was detected
0x8A	Shock detected	Shock was detected
0xA0	Geofence exit zone	Leave geofence zone
0XC0	Geofence entry zone	Enter geofence zone
0x84	Start	Movement started or IN1 or IN2 is detected
0x80	Stop	Movement stop and not IN1 and not IN2 is detected
0x10	Cyclic	Message sent regularly when the EG-IoT is in movement
0x11	Keep Alive	Message sent regularly when the EG-IoT is in rest state

Each transmission takes place in different phases:



In case of Keep Alive, the last available GNSS position is used only if no movement has been detected since the last available position. If the movement detection is disabled, the new GNSS position is obtained each time.

2.1.1 Automatic date and time update process

At each transmission, the application will update the date and time in UTC from the GNSS if it is available.
To overcome this limitation, it is possible to activate the NITZ function which uses the date and time provided by the cellular network for updating the clock of the device in the case of the use of this method of transmission. This feature can be activated from the device configuration application "Ease_EG_IoT_Config" (please contact

Ercogener to obtain this tool). Be careful, the NITZ function is operator and cellular network dependant. In some cases, this function is not provided by the network and cannot be used.

2.2 Special case of operation

When a zone is programmed in geofencing and is located far from the initialisation position of the device, there will be an exit zone frame (opcode A0) during the wake up which will follow the initialisation frame of the device.

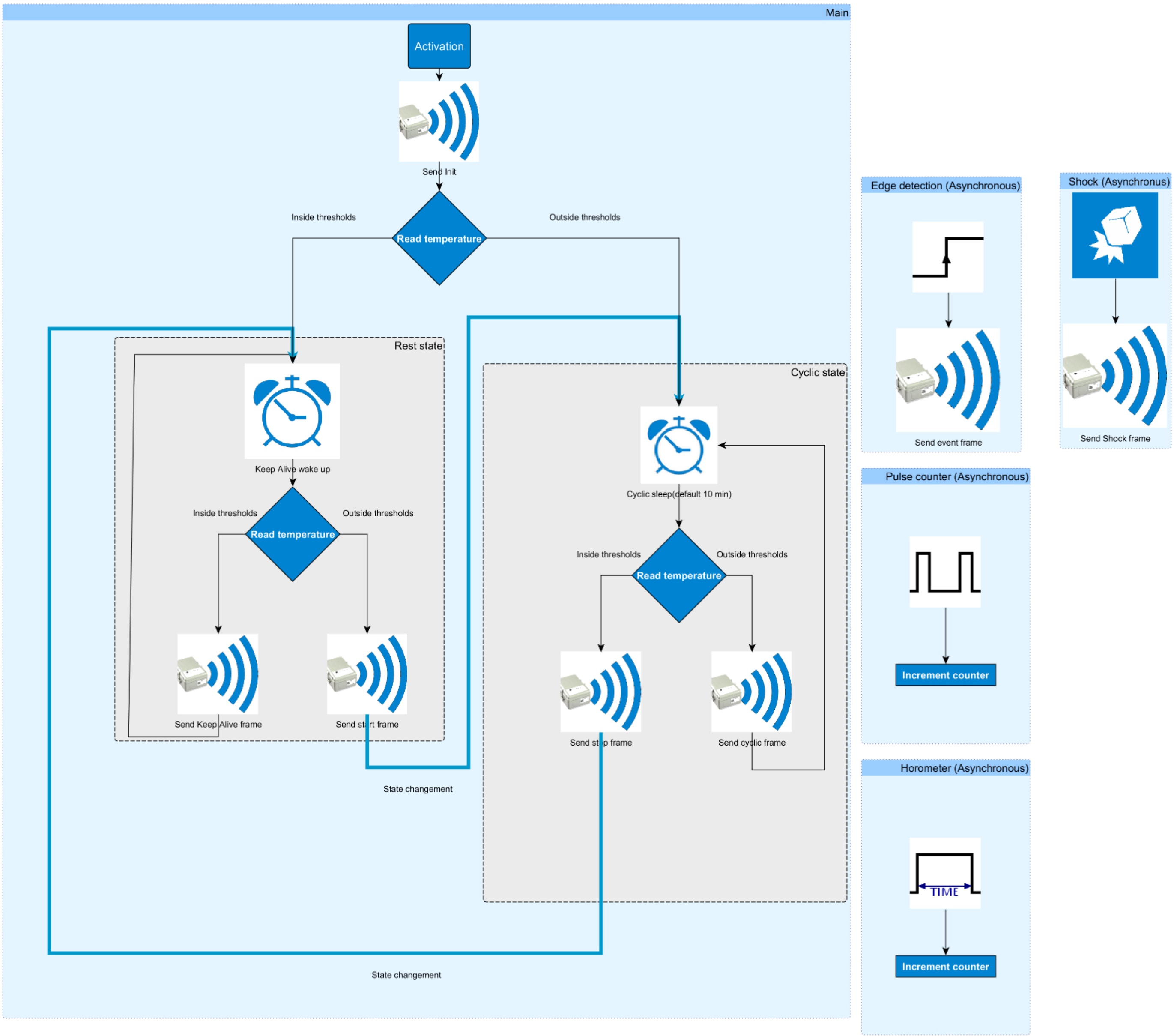
3 Application in remote reading mode

3.1 Operating timing and frame types

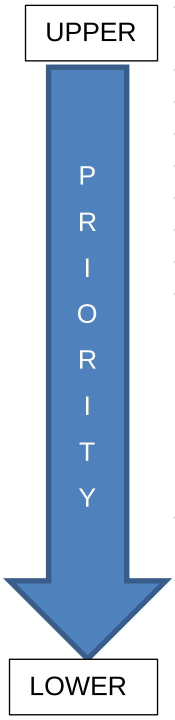
In remote reading mode, the device sends inputs state and other information regularly.
We can distinguish 2 different functioning modes, rest state and temperature alarm state.

In rest state, the device sends frame at the keep alive frequency.
In cyclic state, the device sends frame at the cyclic frequency

The edge detection, pulse counter, horometer and shock detection are independent of the state.

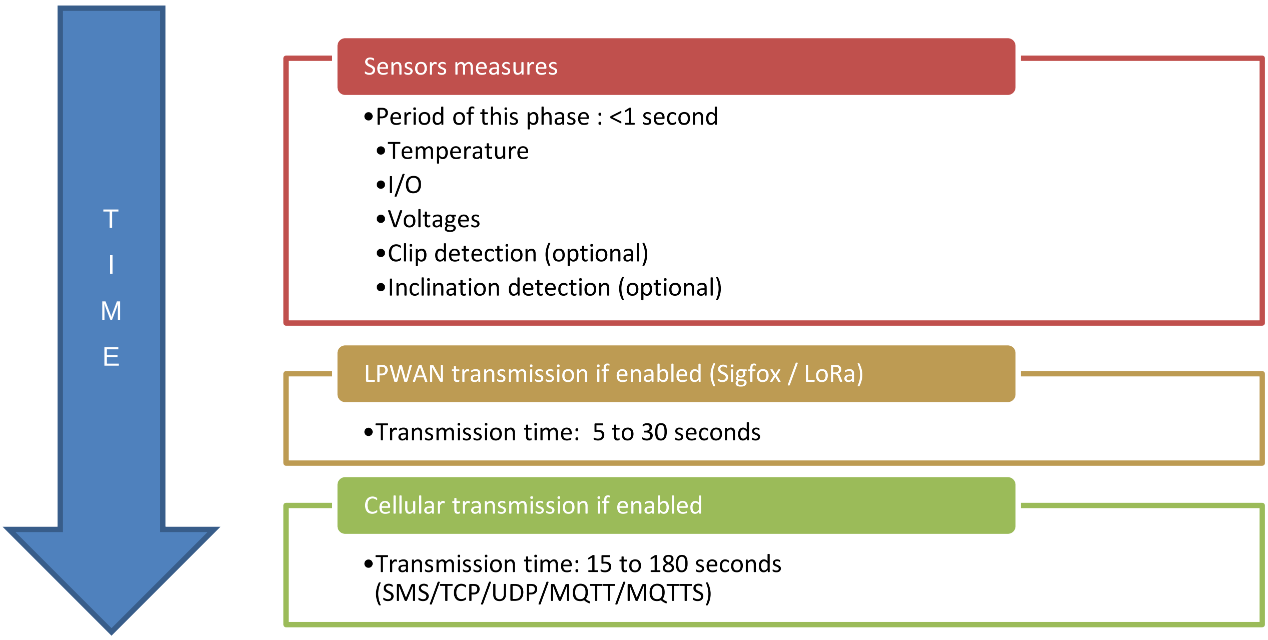


The application can send several types of frame. Each frame has a field called Opcode and its value depends on the functional state.



Opcode	Label	Details
0xFE	Initialisation	Indicates that the device was activated or restarted
0xFF	Exit service mode	Stop application
0x02	LPWAN reset	Reset due to LPWAN downlink
0X86	Angular deviation	Inclination was detected
0x8A	Shock detected	Shock was detected
0x84	Start	Temperature goes outside thresholds for the fist time
0x80	Stop	Temperature goes inside thresholds for the first time
0x17	Event	Message sent if : - Edge detected and entry is not in its rest state - Pulse counter value overlap threshold Horometer value overlap threshold ⚠Upper priority than 0xFF (Exit service mode)
0x10	Cyclic	Message sent regularly when the EG-IoT temperature is outside threshold
0x11	Keep Alive	Message sent regularly when the EG-IoT is in rest state Or edge detected and entry is in its rest state

Each transmission takes place in different phases:



The temperature is not verified and the over temperature detection information is not available in the initialisation frame.

3.2 Automatic date and time update process

In the remote reading mode, the UTC update can only be done from the date and time provided by the cellular network if the NITZ option is enabled and if sending data through this transmission mode is used. This feature can be activated from the device configuration application "Ease_EG_IoT_Config" (please contact **Ercogener** to obtain this tool). Be careful, the NITZ function is operator and cellular network dependent. In some cases, this function is not provided by the network and cannot be used.

4 Additional information

4.1 Transmissions

The frequency of sending frames depends on the communication technology used (due to legal constraints):

Technology	Minimum interval between two sending
Cellular	15 seconds to 3 minutes (Network dependent)
LoRa	1 minute
Sigfox	10 minutes

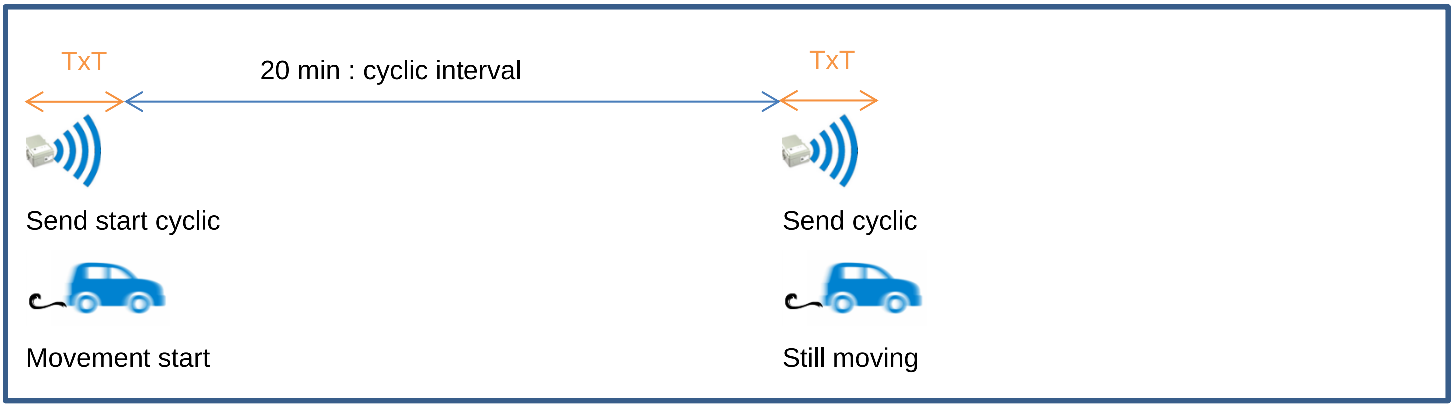
If the EG-IoT must send a message and the communication technology does not allow the frame to be sent immediately, it will retransmit it as soon as it can.

If several messages must be sent at the same time, only that which has upper priority will be sent.

All interval timers are reset at the end of each transmission.

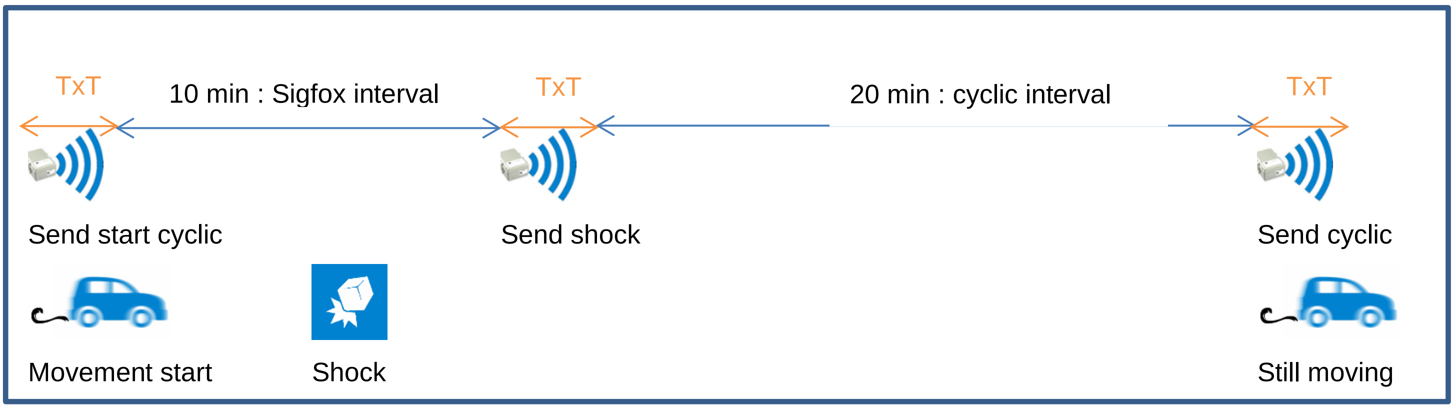
For example:

Sigfox Tracking with cyclic 20 min and shock enable



TxT : Transmission Time

If a shock occurs at T0+5mn, the EG-IoT waits for end of Sigfox interval before sending message



TxT : Transmission Time

4.2 Battery

Two versions are available:

4.2.1 Primary battery (not rechargeable)

The EG-IoT provides an estimation of energy consumed.
Use the remaining battery estimation to determine when battery replacement is required.

After primary battery replacement please reset % bat level to 100%. Use Ease-EG-IoT Config or cellular downlink. See §5.5.1.4.

4.2.2 Secondary battery (rechargeable)

EG-IoT provides product voltage: external voltage if connected or battery voltage.
Monitor battery voltage to determine when battery charging is required.

Do not use the device when battery is under minimum voltage.

The remaining battery capacity field (in %) is not applicable for the secondary battery. It shall not be used. The voltage level shall be used as information of the remaining battery capacity.

4.2.3 Thresholds

Threshold for battery charging or replacement depend on communication technologies.

	LPWAN	Cellular
Primary battery minimum remaining capacity	8%	10%
Secondary battery minimum voltage	3100 mV	3300mV

4.3 Input TOR settings

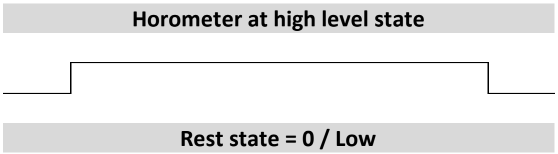
Each input has its own settings:

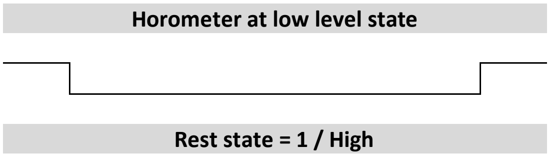
Mode	• - Disabled - Pulse counter (only available in remote reading mode) - Horameter (only available in remote reading mode) - Edge detection
Hold time	• 10 ms to 10 sec (default 60 ms) Useful for filtering bounces
Counter value	• Current counter value • Pulse counter : number of pulses Horameter : seconds ellapsed input not in rest state Edge detection : number of edges detected
Counter threshold	• If a counter value is above threshold send event instead of keep alive frame. To disable this function set to 0 (default value)
Rest state	• 0 (Low) or 1 (High)

4.3.1 Pulse Counter

The counter value is sent at each ***Keep Alive Time***.

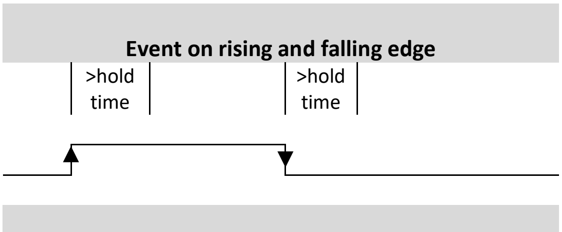
4.3.2 Horometer





The Horometer value is sent at each ***Keep Alive Time***.

4.3.3 Edge detection



At each level change the counter value is incremented.

4.4 Local access via internal serial link

To configure and update the product use the specific configuration tool "**Ease_EG-IoT_Config**" for Windows™ environment. This tool provides a quick configuration of the embedded application "**Ease_EG-IoT**" without any knowledge of the parameters set. Using this software allows access to advanced configuration and operation modes. To obtain it, please contact **Ercogener** support. See documentation "EG_1152_Ease_EG_IoT_Config_UG".

4.5 BLE

The EG-IoT with BLE module can be configured as followed:

- Send advertising frames: EG-IoT is in beacon mode
- Sniffing surrounding beacon tags: EG-IoT is in BLE gateway mode
- Disable: BLE is disabled

4.5.1 Broadcasting mode

In this mode, EG-IoT is considered like a BLE tag.

It sends advertising and scanning frames.

The messages are in hexadecimal format.

The BLE message consists of 3 bytes sent automatically by BLE component in advertising message only (0x020106)
02 nb bytes; 01 Flag & service, data bit 1 LE general discoverable mode, ; 06 data : Bit 1 LE General Discoverable mode / Bit 2 BR/EDR Not supported
These 3 bytes are followed by 3 to 28 characters.

For example 020A04. Or 02 bytes which are type 0A and data 04.
Which means "Tx power level 4dBm"
The types are defined by: ***https://www.bluetooth.com/specifications/assigned-numbers/generic-access-profile/***

The following parameters are available, depending on EG-IoT configuration, the following parameters can be included in advertising or scanning messages and updated after a keep alive or an alert cyclic.

Description	Size (bytes)	Letter	
BLE MAC ID	6	i	#6j
Tx power	1	f	#1f
Latitude	3 or 4	a	#3a or #4a
Longitude	3 or 4	o	#3o or #4o
Altitude	2	A	#2A
Battery capacity	1 (0 to 99%)	n	#1n
Battery voltage	1 (deci Volt) or 2 (milli volt)	v	#1v
Inputs values	1	p	#1p
Input 0	1	J	#1J
Input 1	1	K	#1K
Analog input 0	2	0 (ZERO)	#20
Analog input 1	2	1	#21
Internal temperature	1	z	#1z
Counter 0	2	B	#2B
Counter 1	2	C	#2C

Example of power transmission with Tx power level type:

- message to configure: 020A#1f
- If power is 4dBm, the BLE message sent will be: 020A04

XX 16 30 31 32 34 #6i #4a #4o, XX will be the user data length
16 => service data type
30313234 => 0124 example of specific user data
#6i => ID BLE with 6 bytes length. (FAEDEF234565 in example below)
#4a => latitude 4 bytes length. (123454565654 in example below)
#4o => longitude 4 bytes length. (30313234 in example below)

XX calculation: 1 (type 16) + 4 (0124) + 6 (ID BLE) + 4 (latitude) + 4 (longitude) => 19 bytes (0x13)

The message to configure needs to be: 1316#6i#4a#4o
An example of message sent by device (advertising): 020106131630313234FAEDEF234565123454565654341 2

4.5.2 BLE gateway mode

EG-IoT sniffs BLE tags. For each of them, advertising, scanning messages and RSSI level are collected.

A maximum of 50 filtered tags can be read.

As described in Ease_EG-IoT_Config user guide, BLE tags may be filtered using their names or RSSI levels.

Advertising and scanning frames contents depends on the beacon manufacturer and model.

Example of a Temperature and humidity tag from ELA innovation:

Advertising message : 0x02010605166E2A5E0A004166F2A3010094250555434B53543830304131324E41

Raw data:
0x02010605166E2A5E0A004166F2A3010094
25055434B53543830304131324E41

LEN.	TYPE	VALUE
2	0x01	0x06
5	0x16	0x6E2A5E0A004166F2A30
11	0x09	0x55434B53543830304131324E41

Name (ASCII)

0x6E2A : *Temperature* service
0x6F2A : *Humidity* service

RH data:
- 0x30 : RH data i.e. 48% relative humidity

T° data:
- 0x5E : LSB
- 0x0A : MSB

T° = 0A5E = 2654 * 0.01 = 26.54°C

Note: For a negative temperature, data is sent in 2-complement

BLE sniffing is only available in cellular mode in JSON format

BLE sniffing is done before each EG-IoT transmission (Keep alive, cyclic, shock, …)

BLE sniffing information is not communicated by the EG-IoT as an additional frame to the EG-IoT tracking or remote reading frame.

4.6 Password management

Access to EG-IoT configuration can be protected by an optional login and a password.

By default the login/password is disabled.

The login must contain 4 to 8 ASCII printable characters. The password must contain 8 ASCII printable characters.

The apostrophe (') and quotation marks (") characters are not allowed.

Login / password may be changed up to 30 times during the life of the product.
For the last time the last login/password will be kept.

If login or password has been lost, then there is no way to recover it.

Please provide your login/password when returning your EG-IoT to the factory.

Once you have an active login / password it cannot be disabled.

5 Advanced use

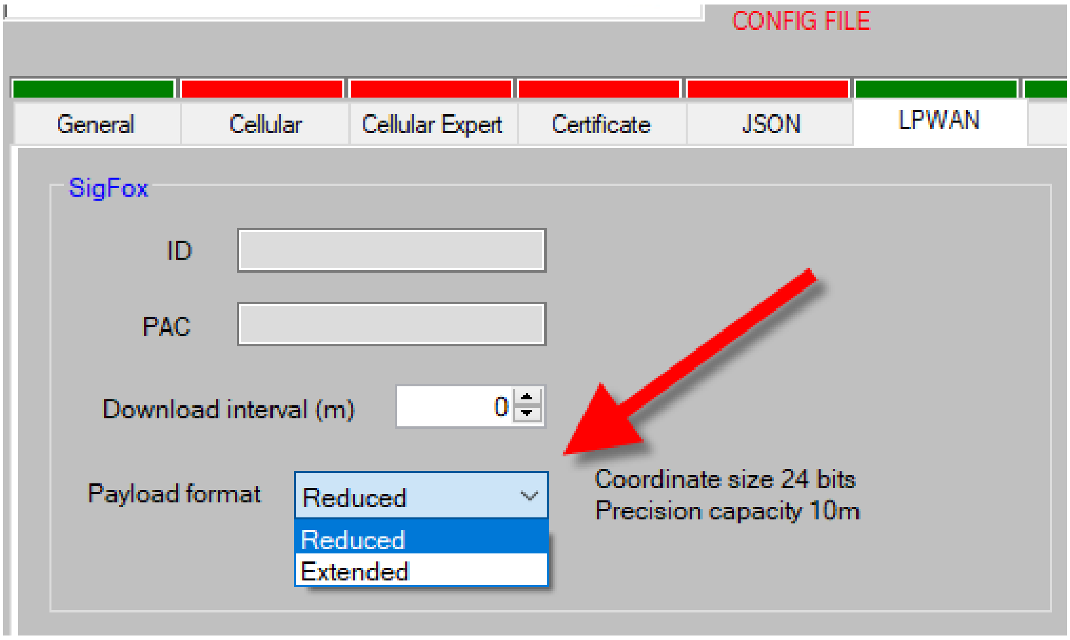
5.1 Payload

5.1.1 Tracking

5.1.1.1 Geolocation: Payload for Sigfox network or LoRa network

The payloads sent by the device have two formats for tracking. There is a format for better coordinate accuracy, extended format, and reduced size.

To use the LPWAN extended payload, use Ease_EG-IoT_Config on LPWAN tab.



5.1.1.2 Tracking: LPWAN Additional payloads information

Battery voltage
Measurement range : 2.00V to 4.55V

Temperature
Measurement range : -40°C to +86°C
Read value 0x7E → 0x7E - 0x28 = 0x56 → 86 degrees
Read value 0x00 → 0x00 - 0x28 = 0xFFFFFFFFFFFFFFD8 → -40 degrees.

The speed unit is 10 km/h
Measurement range : 0 to 500 km/h
30 km/h corresponds to 0x03 hexadecimal (00000011 binary) corresponding to 3 x 10 = 30 km/h
38 km/h also corresponds to 0x03 because of steps of 10 km/h.
40 km/h corresponds to 0x04 hexadecimal (00000100 binary) corresponding to 4 x 10 = 40 km/h

5.1.1.3 *Tracking: LPWAN reduced accuracy payload (LPWAN default)*

With this payload the minimum resolution of the position is 11 m.

SIGFOX/LORA		Opcode	Latitude	Longitude	Reserved	Reserved	V_{BAT}	GNSS Fix	T° int.	Speed 10 km/h	Accuracy	Input state 2	Input state 1
Bits	96	8	24	24	8	8	8	1	7	4	2	1	1
		b7-b0	b23-b0	b23-b0	b7-b0	b7-b0	b7-b0	b7	b6-b0	b7-b4	b3-b2	b1	b0
Bytes	12	0	1 2 3	4 5 6	7	8	9	10		11			
Value		xx											

Opcode	
0x00	Initialisation
0x10	Cyclic
0x11	Keep alive
0x80	Movement stop
0x84	Movement start
0xA0	Exit zone 0
0xC0	Entry zone 0
0xFF	Dummy

GNSS Fix	
Invalid measure	0
Valid measure	1

T° int.	
> 86 °C	0x7F
86 °C	0x7E
0 °C	0x28
-40 °C	0x00

Speed	
0000	Speed
1111	10 Km/h par bit

Accuracy	Accuracy HDOP	Accuracy in meter
00	0<2 Very good	0.0 < 20.0
01	2<5 Good	20.0 <50.0
10	5<9 Moderate	50.0 < 100.0
11	9<20 Poor	100.0 < 65535.0

<u>Example of frame:</u> **10073433FFFDA4000082C630** (hexadecimal value of useful data (*Payload*) Sigfox/LoRa)

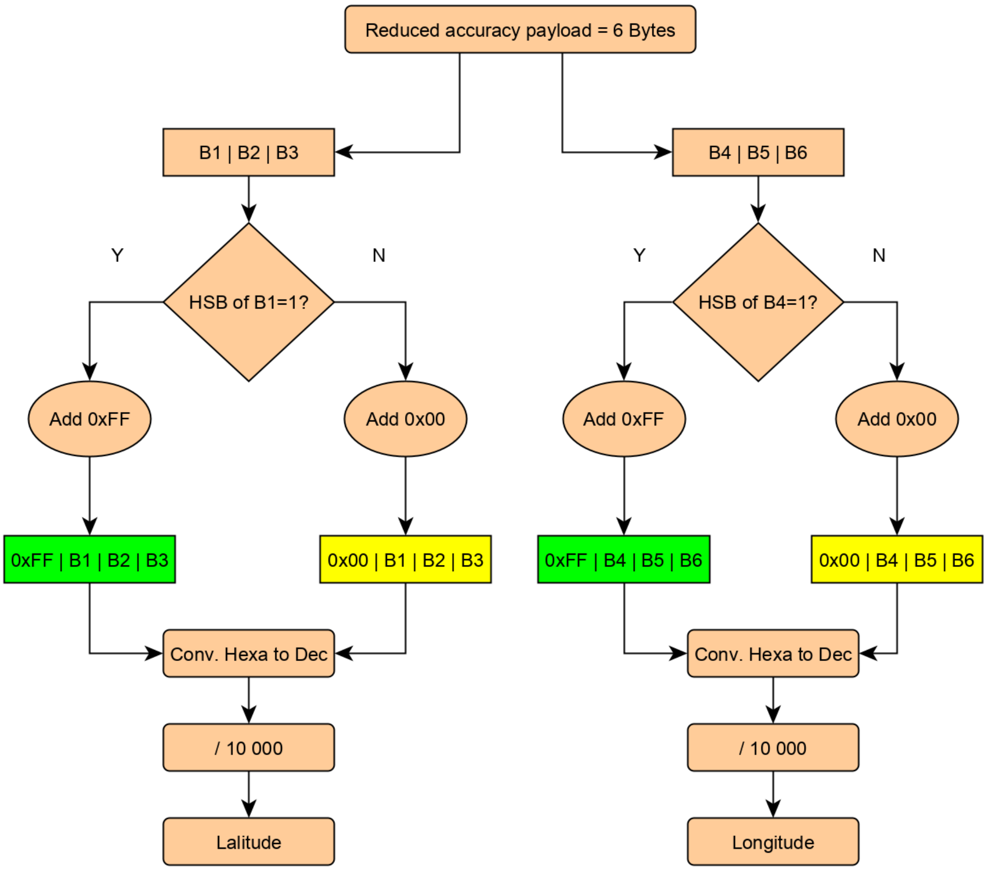
Hexadecimal frame byte sequence: **0x10 0x07 0x34 0x33 0xFF 0xFD 0xA4 0x00 0x00 0x82 0xC6 0x30**

0x10	➔	Opcode	cyclic
0x**073433**	➔	Latitude : **472 115** / 10000	47.2115
0x**FFFDA4**	➔	Longitude : **-604** / 10000	-0.0604
0x**0000**	➔	Reserved	0
0x**82**	➔	Battery voltage: (**130** + 200) / 100	3.30V
0x**C6**	➔	GNSS Fix : **1**	Valid measure
	➔	Internal temperature (0x46): **70** - 40	30°C
0x**30**	➔	Speed : **3** * 10	30Km/h <= Speed < 39Km /h
	➔	HDOP : **0**	Very good
	➔	Input 2: **0**	0
	➔	Input 1: **0**	0

To decode latitude and longitude, these hexadecimal values are 3-byte values and can cause possible decoding errors. You must complete the value to reach 4 bytes by adding the most significant byte of 0xFF or 0x00 ①, convert result to decimal②, then divide the result by 10000③ as follows:

- In the value received 3-byte value, if the most significant bit is 0 then add 0x00:
  - 0x073433 =①> 0x**00**073433 =②> 472115 =③> 47.2115
- In the value received 3-byte value, if the most significant bit is 1 (negative value) then add 0xFF:
  - 0xFFFDA4 =①> 0x**FF**FFFDA4=②> -604 =③> -0.0604

In comparison with previous SW release V2.11, in blue, Input state 1 and 2 replace reserved bits.


Reduced accuracy payload = 6 Bytes
B1 | B2 | B3
B4 | B5 | B6
Y
N
HSB of B1=1?
Y
N
HSB of B4=1?
Add 0xFF
Add 0x00
Add 0xFF
Add 0x00
0xFF | B1 | B2 | B3
0x00 | B1 | B2 | B3
0xFF | B4 | B5 | B6
0x00 | B4 | B5 | B6
Conv. Hexa to Dec
Conv. Hexa to Dec
/ 10 000
/ 10 000
Lalitude
Longitude

5.1.1.4 *Tracking: LPWAN extended accuracy payload*

To get more accuracy (< 11m) on GNSS position it's necessary to use extended accuracy payload.

SIGFOX/LORA		Opcode	Latitude	Longitude	V_{BAT}	GNSS Fix	T° int.	Speed 10 km/h	Accuracy	Input state 2	Input state 1
Bits	96	8	32	32	8	1	7	4	2	1	1
		b7-b0	b31-b0	b31-b0	b7-b0	b7	b6-b0	b7-b4	b3-b2	b1	b0
Bytes	12	0	1 2 3 4	5 6 7 8	9	10		11			
Value		xx									

Opcode	
0x00	Initialisation
0x10	Cyclic
0x11	Keep alive
0x80	Movement stop
0x84	Movement start
0xA0	Exit zone 0
0xC0	Entry zone 0
0xFF	Dummy

GNSS Fix	
Invalid measure	0
Valid measure	1

T° int.	
> 86 °C	0x7F
86 °C	0x7E
0 °C	0x28
-40 °C	0x00

Speed	
0000	Speed
1111	10 Km/h par bit

	Accuracy HDOP	Accuracy in meter
00	0<2 Very good	0.0 < 20.0
01	2<5 Good	20.0 <50.0
10	5<9 Moderate	50.0 < 100.0
11	9<20 Poor	100.0 < 65535.0

Example of frame: **10004809FBFFFFE86C82C630** (hexadecimal value of useful data (*Payload*) Sigfox/LoRa)

Hexadecimal frame byte sequence: **0x10 0x00 0x48 0x09 0xFB 0xFF 0xFF 0xE8 0x6C 0x82 0xC6 0x30**

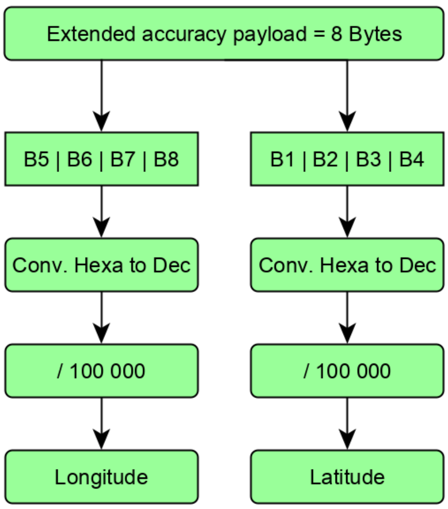
0x**10**	➔	Opcode	cyclic
0x**004809FB**	➔	Latitude **4721147** / 100000	47.21147
0x**FFFFE86C**	➔	Longitude **-6036** / 100000	-0.06036
0x**82**	➔	Battery voltage: (**130** + 200) / 100	3.30V
0x**C6**	➔	GNSS Fix : **1**	Valid measure
	➔	Internal temperature (0x46): **70** – 40	30°C
0x**30**	➔	Speed : **3** * 10	30Km/h <= Speed < 39Km /h
	➔	HDOP : **0**	Very good
	➔	Input 2: **0**	0
	➔	Input 1: **0**	0

To decode latitude and longitude, these are 4-byte hexadecimal values.

Convert result to decimal ① then divide the result by 100000② as follows:

- 0x**004809FB** =①> **4721147** =②> 47.21147
- 0x**FFFFE86C** =①> **-6036** =②> -0.06036

In comparison with previous SW release V2.11, in blue, Input state 1 and 2 replace reserved bits.



5.1.1.5 Tracking: Frame sent by SMS

The transmission of information by SMS indicates the IMEI of the product, the OpCode, the date and time and additional useful information. The link at the end of the SMS allows the location of the product directly in Google Maps©.

SMS		Device ID	Opcode	DATE	TIME	LAT	LONG	Speed	GNSS Fix status	Accuracy HDOP meter	ADC 2	ADC 1	Vbat	T° int	Input 2 state	Input 1 state	Remaining bat	Google Maps link to see the position	GSM RSSI RSRP	GSM Qual RSRQ
Characters	143	15	2	6	6	10	11	5	1	5	6	6	4	3	1	1	2	52	4	3
Separators (,)	18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Total chars	161																			

SMS	Min	Max	Example
Device identifier (default IMEI)	0 chars	15 chars	123456789012345
Opcode (Hexa decimal)	00	C0	10
DATE (ddmmyy)	010119	311299	160419
TIME (hhmmss)	000000	235959	093653
LAT (Decimal degrees)	-89.999999	89.999999	47.211467
LONG (Decimal degrees)	-179.999999	179.999999	-0.060360
Speed (km/h)	0.0	999.9	35.2
GNSS Fix status : 0 : Invalid 1 : GPS Fix (2D/3D) 2 : DGPS Fix	0	2	1
Accuracy HDOP **Accuracy in meter**	0.00 2	99.99 100	2.34 10
Analog input 2 (mV)	-15000	15000	4620
Analog input 1 (mV)	-15000	15000	-3752
Vbat (mV)	0	9999	3300
Tint (°C)	-40	86	25
Input 2 state	0	1	0
Input 1 state	0	1	1
Remaining bat capacity (%)	0	99	53
Google Maps link to see the position	52	52	"https://google.com/maps/place/-89.211467,-160.060360"
GSM RSSI level* or **RSRP (for LTE, valeur in dBm)***	0 -141	31 or 99 -44 or 255	10 -90
GSM QUAL* or **RSRQ (for LTE, valeur in dB)***	0 -19	7 or 99 -3 or 255	10 -9

00 = Initialisation
10 = Cyclic
11 = Keep alive
84 = Movement start
80 = Movement stop
A0 = Exit zone 0
C0 = Entry zone 0
FF = Dummy

99/255 unavailable

In comparison with previous SW release V2.11, in blue, Speed parameter has not changed, its value never exceeds 5 characters. Analog input 1 and 2, Input state 1 and 2 replace reserved parameters. Device identifier can be used instead of IMEI.

Example of SMS 1:

1234564798012345,10,250719,103018,47.211467,-0.060360,35.2,1,1.75,0,0,3531,27,0,0,53,

"**https://www.google.com/maps/place/47.211467,-0.060360"**,10,10

1234564798012345	➔	IMEI	1234564798012345
10	➔	Opcode	Cyclic
250719	➔	Date	25/07/2019
103018	➔	Hour	10H30 and 18s
47.211467	➔	Latitude	47.211467
-0.060360	➔	Longitude	-0.060360
35.2	➔	Speed	35.2 Km/h
1	➔	GNSS status	Valid, GPS Fix (2D/3D)
1.75	➔	HDOP	1.75
0	➔	Analog input 2	0
0	➔	Analog input 1	0
3531	➔	Battery voltage	3.531 V
27	➔	Internal temperature	27°C
0	➔	Digital input 2	0
0	➔	Digital input 1	0
53	➔	Remaining battery capacity	53%
https[…] 0.060360	➔	Google maps link	"**https://www.google.com/maps/place/47.211467,-0.060360"**
10	➔	GSM RSSI	10
10	➔	GSM QUAL	10

Example of SMS 2 :

1234564798012345,10, 250719,103018,,,,0,99.99,0,0,3531,27,0,0,53,10,10

This example of SMS represents the same frame as the first SMS but without a valid GPS position.

1234564798012345	➔	IMEI	1234564798012345
10	➔	Opcode	Cyclic
250719	➔	Date	25/07/2019
103018	➔	Hour	10H30 and8s
	➔	Latitude	
	➔	Longitude	
	➔	Speed	
0	➔	Status GNSS	Invalid
99.99	➔	HDOP	99.99
0	➔	Analog input 2	0
0	➔	Analog input 1	0
3531	➔	Battery voltage	3.531 V
27	➔	Internal temperature	27°C
0	➔	Digital input 2	0
0	➔	Digital input 1	0
53	➔	Remaining battery capacity	53%
	➔	Google maps link	
10	➔	GSM RSSI	10
10	➔	GSM QUAL	10

WARNING, in the case where the GNSS position is invalid, google maps link is missing and it lacks a separator between the remaining battery capacity and cellular RSSI / RSRP.

5.1.1.6 Tracking: Frame sent via TCP or UDP or MQTT

The transmission of information indicates the IMEI of the product, the OpCode, the date and time and additional useful information.

TCP/UDP		Device ID	Opcode	DATE	TIME	LAT	LONG	Speed	GNSS Fix status	Accuracy HDOP meter	ADC 2	ADC 1	Vbat	T° int	Input 2 state	Input 1 state	Altitude	Remaining bat	GSM RSSI RSRP	GSM Qual RSRQ
Characters	98	15	2	6	6	10	11	6	1	5	6	6	4	3	1	1	6	2	4	3
Separators (,)	18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Total chars	116																			

TCP/UDP	Min	Max	Exemple
Device identifier (default IMEI)	0 chars	15 chars	123456789012345
Opcode (Hexa decimal)	**00**	**C0**	**10**
DATE (ddmmyy)	010119	311299	160419
TIME (hhmmss)	000000	235959	093653
LAT (Decimal degrees)	-89.999999	89.999999	47.211467
LONG (Decimal degrees)	-179.999999	179.999999	-0.06036
Speed (km/h)	0.00	999.99	35.20
GNSS Fix status : 0 : Invalid 1 : GPS Fix (2D/3D) 2 : DGPS Fix	0	2	1
Accuracy HDOP **Accuracy in meter**	0.00 2	99.99 100	2.34 10
Analog input 2 (mV)	-15000	15000	4620
Analog input 1 (mV)	-15000	15000	-3752
Vbat (mV)	0	9999	3300
Tint (°C)	-40	86	25
Input 2 state	0	1	0
Input 1 state	0	1	1
Altitude (m)	-12000	12000	120
Remaining bat capacity (%)	0	99	53
Cellular RSSI level* or **RSRP (for LTE, valeur en dBm)***	0 -141	31 ou 99 -44 ou 255	10 -90
Cellular QUAL* or **RSRQ (for LTE, valeur in dB)***	0 -19	7 or 99 -3 or 255	10 -9

00 = Initialisation
10 = Cyclic
11 = Keep alive
84 = Movement start
80 = Movement stop
A0 = Exit zone 0
C0 = Entry zone 0
FF = Dummy

99/255 unavailable

* see ANNEX for more information

In comparison with previous SW release V2.11, in blue, Analog input 1 and 2, Input state 1 and 2 replace reserved parameters. Device identifier can be used instead of IMEI.

Example of frame:

1234564798012345,10,250719,103018,47.211467,-0.060360,35.20,1,1.75,0,0,3531,27,0,0,108,53,10,10

1234564798012345	→	IMEI	1234564798012345
10	→	Opcode	Cyclic
250719	→	Date	25/07/2019
103018	→	Hour	10H30 and 18s
47.211467	→	Latitude	47.211467
-0.060360	→	Longitude	-0.060360
35.20	→	Speed	35.2 Km/h
1	→	GNSS status	Valid, GPS Fix (2D/3D)
1.75	→	HDOP	1.75
0	→	Analog input 2	0
0	→	Analog input 1	0
3531	→	Battery voltage	3.531 V
27	→	Internal temperature	27°C
0	→	Digital input 2	0
0	→	Digital input 1	0
108	→	Altitude	108m
53	→	Remaining battery capacity	53%
10	→	Cellular RSSI/RSRP	10
10	→	Cellular QUAL/RSRQ	10

5.1.2 Remote reading:

5.1.2.1 Remote reading: LPWAN payload

SIGFOX/LORA		Opcode	Digital input 1 counter		Digital input 2 counter		ADC 2		ADC 1		Remaining bat	V_{BAT}	X	T° int.	Reserved	Output state	Temperature high threshold	Temperature low threshold	Reserved	Input 2 state	Input 1 state
Bit	96	8	16		16		12		12		8	8	1	7	1	1	1	1	2	1	1
		b7-b0	b15-b0		b15-b0		b7-b0	b7-b4	b3-b0	b7-b0	b7-b0	b7-b0	b7	b6-b0	b7	b6	b5	b4	b3-b2	b1	b0
Octect	12	0	1	2	3	4	5	6		7	8	9	10		11						
Valeur		xx												xx							

Opcode	
0xFE	Initialisation
0x11	Keep alive
0x17	Event
0xFF	Dummy

ADC 2 / ADC 1: Analog input value 10mv by step

T° int.	
0x7F	> 86°C
0x7E	86°C
0x29	0°C
0x00	-40°C

Temperature low threshold	
0	°C > low threshold
1	°C <= low threshold

Temperature high threshold	
0	°C > high threshold
1	°C <= high threshold

In comparison with previous SW release V2.11, in blue, Analog input 1 and 2 replace 16 bits reserved but we add 8 bits for these, output state replaces reserved bit.

Example of frame: **110064002000000035824642** (hexadecimal value of useful data (*Payload*) Sigfox/LoRa)
Hexadecimal frame byte sequence: **0x11 0x00 0x64 0x00 0x20 0x00 0x00 0x00 0x35 0x82 0x46 0x42**

0x**11**	➔	Opcode	**Keep alive**
0x**0064**	➔	Digital input 1 counter	**100**
0x**0020**	➔	Digital input 2 counter	**32**
0x**000**	➔	Analog input 2	0
0x**000**	➔	Analog input 1	0
0x**35**	➔	Remaining battery capacity	53 %
0x**82**	➔	Battery voltage : (**130** + 200) / 100	3.30V
0x**46**	➔	X : **0**	0
	➔	Internal temperature: **70** - 40	30°C
0x**42**	➔	Reserved: 0	0
	➔	Output state	1
	➔	Temperature high threshold : 0	0, no exceeding high temperature
	➔	Temperature low threshold : 0	0, no exceeding low temperature
	➔	Reserved: 00	00
	➔	Input 2 state : 1	1, input 2 at high level
	➔	Input 1 state : 0	0, input 1 at low level

Battery voltage
Measurement range: 2.00V to 4.55V

Temperature
Measurement range: -40°C to +86°C
Read value 0x7E → 0x7E - 0x28 = 0x56 → 86 degrees
Read value 0x00 → 0x00 - 0x28 = 0xFFFFFFFFFFFFFFD8 → -40 degrees.

5.1.2.2 Remote reading: SMS payload

SMS		Device ID	Opcode	DATE	TIME	Counter 1 value	Counter 2 value	Counters status	reserved	Remaining bat	Vbat	T° int	Output state	Input 1 state	Input 2 state	GSM RSSI RSRP	GSM Qual RSRQ	ADC 2	ADC 1
Characters	83	15	2	6	6	10	10	2	1	2	4	3	1	1	1	4	3	6	6
Separators (,)	17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Total chars	100																		

SMS	Min	Max	Example
Device identifier (default IMEI)	000000000000000	999999999999999	123456789012345
Opcode (Hexa decimal)	11	FE	11
Date (ddmmyy)	010119	311299	090519
Heure (hhmmss)	000000	235959	122123
Counter 1 value	0	4 294 967 295	47
Counter 2 value	0	4 294 967 295	259
Counters status	0	FF	02
Reserved	0	0	0
Remaining bat (%)	1	99	53
Vbat (mV)	0	9999	3320
Temperature (°C)	-40	86	25
Output state	0	1	0
Input 1 state	0	1	0
Input 2 state	0	1	1
GSM RSSI level* or **RSRP (for LTE, valeur in dBm)***	0 -141	31 or 99 -44 or 255	10 -90
GSM QUAL* or **RSRQ (for LTE, valeur in dB)***	0 -19	7 or 99 -3 or 255	10 -9
Analog input 2 (mV)	-15000	15000	4620
Analog input 1 (mV)	-15000	15000	-3752

Opcode	
11	Keep alive
17	Event
FE	Initialisation
FF	Dummy

Counters status	
b0	Input 1 state
b1	Input 2 state
b2,b3	Reserved
b4	Low temperature threshold
0	< low threshold
1	≥ low treshold
b5	High temperature threshold
0	< high threshold
1	≥ high threshold
b6	output state
b7	Reserved

***** see ANNEX for more details

In comparison with previous SW release V2.11, in blue, output state replaces reserved parameter, Analog input 1 and 2 are added at the end. Device identifier can be used instead of IMEI.

Example of SMS: **1234564798012345,11,250719,103018,100,32,66,0,53,3531,27,1,0,1,10,10,4620,-3752**

1234564798012345	➔	IMEI	1234564798012345
11	➔	Opcode	Keep alive
250719	➔	Date	25/07/2019
103018	➔	Hour	10H30 and 18s
100	➔	Digital input 1 counter	100
32	➔	Digital input 2 counter	32
66	➔	Reserved: 0	0
	➔	Output state: 1	1
	➔	Temperature high threshold: 0	0, no exceeding high temperature
	➔	Temperature low threshold: 0	0, no exceeding low temperature
	➔	Reserved: 00	00
	➔	Input 2 state: 1	1, input 2 at high level
	➔	Input 1 state: 0	0, input 1 at low level
0	➔	Reserved	0
53	➔	Remaining battery capacity	53 %
3531	➔	Battery voltage	3.531V
27	➔	Internal temperature	27°C
1	➔	Output state: 1	1, output at low level
0	➔	Input 1 state: 0	0, input 1 at low level
1	➔	Input 2 state : 1	1, input 2 at high level
10	➔	Cellular RSSI/RSRP	10
10	➔	Cellular QUAL/RSRQ	10
4620	➔	Analog input 2	4.62 V
-3752	➔	Analog input 1	-3.752V

5.1.2.3 Remote reading: TCP or UDP payload

TCP/UDP		Device ID	Opcode	DATE	TIME	Counter 1 value	Counter 2 value	Counters status	reserved	Remaining bat	Vbat	T° int	Output state	Input 1 state	Input 2 state	GSM RSSI RSRP	GSM Qual RSRQ	ADC 2	ADC 1
Characters	83	15	2	6	6	10	10	2	1	2	4	3	1	1	1	4	3	6	6
Separators (,)	17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Total chars	100																		

TCP/UDP	Min	Max	Example
Device identifier (default IMEI)	000000000000000	999999999999999	123456789012345
Opcode (Hexa decimal)	11	FE	11
Date (ddmmyy)	010119	311299	090519
Heure (hhmmss)	000000	235959	122138
Counter 1 value	0	4 294 967 295	47
Counter 2 value	0	4 294 967 295	259
Counters status	00	FF	02
Reserved	0	0	0
Remaining bat (%)	1	99	53
Vbat (mV)	0	9999	3320
Temperature (°C)	-40	86	25
Output state	0	0	0
Input 1 state	0	1	0
Input 2 state	0	1	1
GSM RSSI level* or **RSRP (for LTE, valeur in dBm)***	0 -141	31 or 99 -44 or 255	10 -90
GSM QUAL* or **RSRQ (for LTE, valeur in dB)***	0 -19	7 or 99 -3 or 255	10 -9
Analog input 2 (mV)	-15000	15000	4620
Analog input 1 (mV)	-15000	15000	-3752

Opcode:

11	Keep alive
17	Event
FE	Initialisation
FF	Dummy

Counters status:

b0	Input 1 state
b1	Input 2 state
b2,b3	Reserved
b4	Low temperature threshold
0	< low threshold
1	≥ low threshold
b5	High temperature threshold
0	< high threshold
1	≥ high threshold
b6	output state
b7	Reserved

***** see ANNEX for more details

In comparison with previous SW release V2.11, in blue, output state replaces reserved parameter, Analog input 1 and 2 are added at the end. Device identifier can be used instead of IMEI.

<u>**Example of frame:**</u> **1234564798012345,11,250719,103018,100,32,66,0,53,3531,27,1,0,1,10,10,4620,-3752**

1234564798012345	➔	IMEI	1234564798012345
11	➔	Opcode	Keep alive
250719	➔	Date	25/07/2019
103018	➔	Hour	10H30 and 18s
100	➔	Digital input 1 counter	100
32	➔	Digital input 2 counter	32
66	➔	Reserved: 0	0
	➔	Output state: 1	1, output at high level
	➔	Temperature high threshold: 0	0, no exceeding high temperature
	➔	Temperature low threshold: 0	0, no exceeding low temperature
	➔	Reserved: 00	00
	➔	Input 2 state: 1	1, input 2 at high level
	➔	Input 1 state: 0	0, input 1 at low level
0	➔	Reserved	0
53	➔	Remaining battery capacity	53 %
3531	➔	Battery voltage	3.531V
27	➔	Internal temperature	27°C
1	➔	Output level	1
0	➔	Input 1 state: 0	0, input 1 at low level
1	➔	Input 2 state : 1	1, input 2 at high level
10	➔	Cellular RSSI/RSRP	10
10	➔	Cellular QUAL/RSRQ	10
4620	➔	Analog input 2	4.62 V
-3752	➔	Analog input 1	-3.752V

5.2 Managed parameters

The table below shows available information in order to choose the most adapted transmission technology for your use case

.

				SIGFOX/LORA											
				fix payload			CELLULAR								
				Reduced size		extended	SMS		TCP/UDP fix payload		TCP/UDP and JSON dynamic payload				
	Type value	Min value	Max value	Tracking	Telemetry	Tracking	Tracking	Telemetry	Tracking	Telemetry	Tracking or Telemetry				
Device identifier	15 alphanumeric digits						X	X	X	X	DEV_ID				
OPCODE	hexadecimal	0x00	0xFF	8 bits	8 bits	8 bits	X	X	X	X	$OPCODE_INT$	$OPCODE_STRING$			
EVENT DATE	ddmmyy	10119	311299				X	X	X	X	$DATE_1$	$DATE_2$	$DATE_3$	$DATE_4$	$DATE_TIME$
EVENT TIME	hhmmss	0	235959				X	X	X	X	$TIME$	$TIMESTAMP$	$DATE_TIME$		
Latitude	Decimal degrees	-89.999999	89.999999	24 bits		32 bits	X		X		LAT				
Longitude	Decimal degrees	-179.999999	179.999999	24 bits		32 bits	X		X		LON				
Speed	km/h	0	999,99	4 bits		4 bits	X		X		$SPEED$				
GNSS FIX	0 : Invalid 1 : GPS Fix (2D/3D) 2 : DGPS Fix	0	2	1 bit		1 bit	X		X		$GNSS_FIX_INT$	$GNSS_FIX_STRING$			
Accuracy HDOP		0.00	99.99	2 bits		2 bits	X		X		$ACCURACY_MODE$	$ACCURACY_STRING$	$ACCURACY_DEC$	$ACCURACY_INT$	
Accuracy in meter	meter	0	65535				X		X		$ACCURACY_MODE$	$ACCURACY_STRING$	$ACCURACY_DEC$	$ACCURACY_INT$	
Analog input 2	mV	-15000	15000		12 bits		X	X	X	X	$ADC1_VALUE$				
Analog input 1	mV	-15000	15000		12 bits		X	X	X	X	$ADC2_VALUE$				
Battery Voltage	mV	0	9999	8 bits	8 bits	8 bits	X	X	X	X	$VBAT_V$	$VBAT_MV$			
Internal Temperature	°C	-40	86	7 bits	7 bits	7 bits	X	X	X	X	$TEMP$				
Input 2 state		0	1	1 bit	1 bit	1 bit	X	X	X	X	$IN2_STATE$				
Input 1 state		0	1	1 bit	1 bit	1 bit	X	X	X	X	$IN1_STATE$				
Altitude	m	-12000	12000						X		ALT				
Remaining bat capacity	%	0	99		8 bits		X	X	X	X	BAT_CAP				
Google Maps link	HTML	(52 characters)	(52 characters)				X								
2G/3G RRSI level		0	31 or 99				X	X	X	X	$SIG_STRENGTH$				
4G RSRP	dBm	-141	-44 or 255				X	X	X	X	$SIG_STRENGTH$				
2G/3G QUAL		0	7 or 99				X	X	X	X	SIG_QUAL				
4G RSRQ	dBm	-19	-3 or 255				X	X	X	X	SIG_QUAL				
Counter 1 value		0	65535		16 bits			X		X	$CPT1_VALUE$				
Counter 2 value		0	65535		16 bits			X		X	$CPT2_VALUE$				
Counter input status	hexadecimal	0x00	0xFF					X		X	$ALARM_HEX$	$ALARM_INT$			
Alert level (threshold)		0x00	0xFF		2 bits			X		X					
Output state		0	1		1 bit			X		X	OUT				
IMEI	15 numeric digits										$IMEI$				

				SIGFOX/LORA											
				fix payload			CELLULAR								
				Reduced size		extended	SMS		TCP/UDP fix payload		TCP/UDP and JSON dynamic payload				
	Type value	Min value	Max value	Tracking	Telemetry	Tracking	Tracking	Telemetry	Tracking	Telemetry	Tracking or Telemetry				
LPWAN Module Identifier (LORA)	16 hexadecimal digits										$LPWAN_ID$				
Hardware capacity of the EG-IoT device	alphanumeric string										$HW_CAPACITY$				
Radio access technology used for the current cellular connection	0: GSM 2: UMTS 3: LTE 7: LTE-M 8: NB-IoT 9: GPRS-eGPRS	0	9								$URAT$				
Cell identifier	hexadecimal	0x00000000	0xFFFFFFFF								$CID0$	$CID1$	$CID2$	$CID3$	$CID4$
Location Area Code	hexadecimal	0x0000	0xFFFF								$LAC0$	$LAC1$	$LAC2$	$LAC3$	$LAC4$
Mobile Country Code	decimal 3 digits	000	999								$MCC0$	$MCC1$	$MCC2$	$MCC3$	$MCC4$
Mobile Network Code	decimal 1 to 3 digit	0	999								$MNC0$	$MNC1$	$MNC2$	$MNC3$	$MNC4$
Radio Access Technology of cell-id	"2G" "3G" "4G"	"2G"	"4G"								$RAT0$	$RAT1$	$RAT2$	$RAT3$	$RAT4$
Signal strength of cell-id	3 digits	0	63/255								$RSSI0$	$RSSI1$	$RSSI2$	$RSSI3$	$RSSI4$
Product signature (functionalities)	hexadecimal 8 bytes										$SIGNATURE$				
Angle axe des X	decimal	0	180								$ANGLE_X$				
Angle axe des Y	decimal	0	180								$ANGLE_Y$				
Angle axe des Z	decimal	0	180								$ANGLE_Z$				
Product can be powered from external supply		0	1								PWR_SUPPLY				
External power supply voltage	decimal	0	30000								$EXT_PWR_SUPPLY_MV$				

5.3 IP transfer

5.3.1 Establishing of the TCP / UDP layer

Once the device has attached to the cellular network, it opens the TDP/UDP socket. If an error occurs whilst establishing the TCP/UDP socket including the DNS resolution, the application will make up to three attempts, with a timeout of 30s for each attempt. The MQTT protocol using a TCP link is impacted by this mechanism.

5.4 SIM management for cellular transmission

The SIM management is useful for SMS and IP transfer, for downlink and uplink. The SIM card may be configured with a PIN code if required. In case of wrong configured PIN code, the device tries to unlock the SIM card once. If the SIM card is SIM locked, then there is no way to unlock the SIM card with Ease_EG-IoT. Another kind of device, or smartphone may be used to unlock it with the PUK code.

5.4.1 Acknowledge of TCP/UDP data

In the cellular configuration, when the TCP or UDP protocol is configured, it is strongly recommended to use the acknowledgment to ensure the correct data feedback.

The use of this acknowledgment mechanism requires the use of a compatible TCP/UDP server. The TCP/UDP server must send to the device a known ASCII string upon receiving a data frame in order to acknowledge it. If the acknowledgment frame, previously configured in the device, is not received by the device after having sent a data frame, then the frame will be re-transmitted 3 times and then the connection will be closed. The frame will be saved in memory and re-sent on the next alarm, as described below:

In the case where the transmission of the data is configured in cellular mode, TCP, UDP or MQTT, and when the transmission of data has failed, then it will be re-sent at the next wake up. In the case where several transmissions have failed, all unsent measurements will be sent when the connection allows. The oldest measurement is sent first up until the most recent measurement. A transmission is also considered as failed if the acknowledge is configured but not received after transmission.

The cellular reception levels contained in the frames are measured when sending each frame and added when recording of the frame.

5.4.2 MQTT transfer error handling

The MQTT protocol is based on the TCP/IP protocol. The establishment of the TCP socket is therefore subject to the same mechanism described above. Once the TCP socket is established, an MQTT connection is executed. If the connection is not acknowledged, a CONNACK frame is not received by the device. A reconnection of the TCP layer is then performed up to three times. In case of non-acknowledgment of the publication of the data, a PUBACK frame is not received by the device. The publication MQTT is carried out up to three times, and then the connection will be closed. The frame will be saved in memory and re-sent on the next alarm.

5.4.3 Cell-id

With Cell-id EG-IoT lists ONLY the 5 first antenna IDs given by the cellular module.

Data can be processed later by remote server to deduce location.

⚠ We do not recommend the simultaneous use of PSM (LTE-M or NB-IoT required) and Cell-ID due to a functional restriction of the cellular module.

⚠ Using cell id reduces EG-IoT autonomy.

⚠ Cell id uses IP data service, beware of using an adapted SIM provider subscription.

In the cellular configuration, enable cell-id, to locate your device with cellular network information. The cell-id may be disabled, enabled if GNSS position is not found or always enabled. This mode is only available with JSON payload format and dedicated JSON structure, as described in annex.

To check Cell-id, see **https://opencellid.org/**.

5.4.4 Cell locate

With cell locate, EG-IoT acquires the identifiers of visible cells, and following exchanges with a dedicated server returns a position.

⚠ Cell locate is available only with 2G and 3G network.

⚠ Using cell locate reduce EG-IoT autonomy.

⚠ Cell locate uses ip data service, beware to use an adapted SIM provider subscription.

In the cellular configuration, enable cell-locate to locate your device with cellular network information. If GNSS position is not found the EG-IoT try to determine her by using cellular network.

5.5 Cellular Downlink

Cellular downlink will be received only if the device wakes up to send frame by cellular link.

5.5.1 Parameter write command

The following commands must be used to change the parameters values.

Programmable value uses a Hayes special register mechanism based on a register value followed by an index, “Sn.m”.

Double quotes shall not be put.

For example S1 is a global register used for general configuration process almost like "General" folder in Ease_EG_ioT_config software. Example: S1.4 is used for “Keep alive” time value.

⚠ Space character is reserved to separate the fields. Do not use it in command string.

⚠ Semi colon character is reserved to concatenate command in one string.

If two parameters that follow each other must be modified, this can be done by a single command using the semicolon separator. → `Sn.m=xx;yy`

If two parameters that do not follow each other must be modified, this can be done by a double command using the space separator. → `Sn.m=xx Sn.n=yy`

5.5.1.1 Date and Time

Menu value:

```
 1: Date ("dd/mm/yyyy")                           [01/01/2020]
 2: Time UTC ("hh:mm"24H)                         [12:00]
```

Remote parameter

```
      S1.1
```

Command

```
      S1.1=06/03/2020            Set date
      S1.2=09:41                 Set time
      S1.1=06/03/2020;09:41      Set date and time
```

5.5.1.2 Keep alive time

Menu value:

```
4: Keep alive time ("I,mn";"F,[0-7],hh:mm") [I,480]
```

Remote parameter

```
      S1.4
```

Command

```
      S1.4=I,5
```

5.5.1.3 Alert cyclic wakeup

Menu value:

```
5: Alert cyclic wakeup ("HH:MM:SS")          [00:10:00]
```

Remote parameter

```
      S1.5
```

Command

```
      S1.5=00:15:00
```

5.5.1.4 Battery capacity

Menu value:

```
 B: Reset estimated remaining battery capacity
```

Remote parameter

```
      S1.8
```

Command

```
      S1.8=100
```

5.5.1.5 Digital output state

Menu value:

```
      '?': Read input state
```

Remote parameter

```
      S15
```

Command

```
      S15=0 to reset output state
      S15=1 to set output state
```

5.5.1.6 Parameter read command

Send a downlink command to read the general product configuration.
The format is:
<Sx?>

Only one configuration can be read at a time.
X value can be only:
1, general parameter
15, output state

The response format depends on selected transmission, SMS, TCP/UDP, JSON.

1	352753093032595	Product identifier
2	0	Mode 0 : Tracking 1 : Remote reading
3	11/09/2019	Product current date
4	10:56	Product current hour
5	99999	The timeout before its activation Available values : • 0 do not activate the product • from 1 to 10079 minutes (7 days) • 99999 immediate activation
6	I,6	The period of transmission for the Keep-Alive frame. Operating modes: • Interval period : "I,X", with X from 6 to 10079 minutes (7 days) • Fixed period : • "F,**0**,HH:MM", every day at HH:MM ⇨ "F,**X**,HH:MM", X from 1 for Monday to 7 for Sunday, at HH:MM
7	00:10:00	The period of transmission of frames in alert mode
8	0	Transmission mode 0 : Cellular 1 : LPWAN 2 : LPWAN + Cellular
9	1	Payload Battery value 1 : Voltage 2 : Capacity
10	98	Percentage of remaining battery
11	1	Cyclic move sent

5.5.2 JSON format for downlink

To use JSON data format in downlink, no minimal configuration. Can use JSON downlink in TCP/UDP and in MQTT.
The JSON object name in configuration is used to define the name of the JSON object containing the value to configure, or empty if the object name matches the **Ercogener** format.

The format use is the same as Hayes special register mechanism. In case of JSON, ‘@’ char is behind SX.Y in JSON value name. Use ‘?’ in JSON object value to read general register.

Additional fields may be present in the JSON structure and will be ignored in downlink. These fields may be used for ACK, adding downlink identifier for example.

Example 1:

- ValueName : "v"

```
{
	"cfg":
	{
		"@S1.4":
		{
			"t": "string",
			"v": "I,15"
		},
		"@S1.5":
		{
			"t": "string",
			"v": "00:10:00"
		},
	},
	"cid": "695232014"
}
```

Example 2:

- ValueName empty

```
{
	"cfg":
	{
		"@S1.4": "I,15",
		"@S1.5": "00:10:00"
	},
	"config_id": "695232014"
}
```

Example 3:

- Read value

```
{
	"cfg":
	{
		"@S1": "?"
	},
	"config_id": "695232014"
}
```

Acknowledge format

If everything is OK, the device sends exactly the same structure as the one it received with the exception of the fields corresponding to a read command. For these fields, the value "?" is replaced by the value of the configuration of the device.

In case of error on command index “@SX.Y”, the value of the command contains: “NOT ALLOWED”

In case of an error on the json structure of the downlink, the response will be:

{"RESULT": "ERR JSON FORMAT"}

In case of a memory error, the response will be:

{"RESULT": "ERR MEMORY"}

If the device does not find a configuration command in the downlink, the response will be:

{"RESULT": "ERR CMD NOT FOUND"}

In the case of an error on a command, its value will be replaced by "ERR VALUE".

⚠ To erase any parameter, send "@Sx.x": "\" \"". The space (“ “) is not accepted.

5.5.3 SMS remote control

To use this feature, the customer should activate this first, using cellular and configuring remote control password in SMS configuration.

This password is not the same as that describe for product access in chapter §5.3.Password Management.

A SMS with a specific structure described in previous chapter is used to send a command to the product.

This message must start with a password followed by one or more commands each separate by space:

```
<PWD> <Cmdx=n> <Cmdy.z=n;m>
```

Space and semi-colon characters are reserved. Do not use them in password.

5.5.3.1 SMS command example

```
<0000 s1.4=I,5;00:15:00>
```

5.5.3.2 SMS Command acknowledgment

When the SMS is handled an acknowledgment is returned.

The format is:

<Product identifier> <Sn>:<OK/KO> <Sn.m>:<OK/KO>
By default the product identifier is the product IMEI.

Example:

```
<352753093012345 s1.4:OK s1.5:OK>
```

In the case where a multiple command is sent (with semi-column separator) then if the response <KO> is returned, it is not possible to know which parameter is incorrect.

5.5.3.3 SMS read command

The SMS read command returns the register value with the name of each parameter.

Example:

```
352753093032595 S1 MD:1; DT:10/03/2020;10:03; AD:99999; KA:I,6; CW:00:10:00; TXM:0; 1; BC:99; CF:0
```

If the answer exceeds the SMS size (160 characters), it will be cut off.

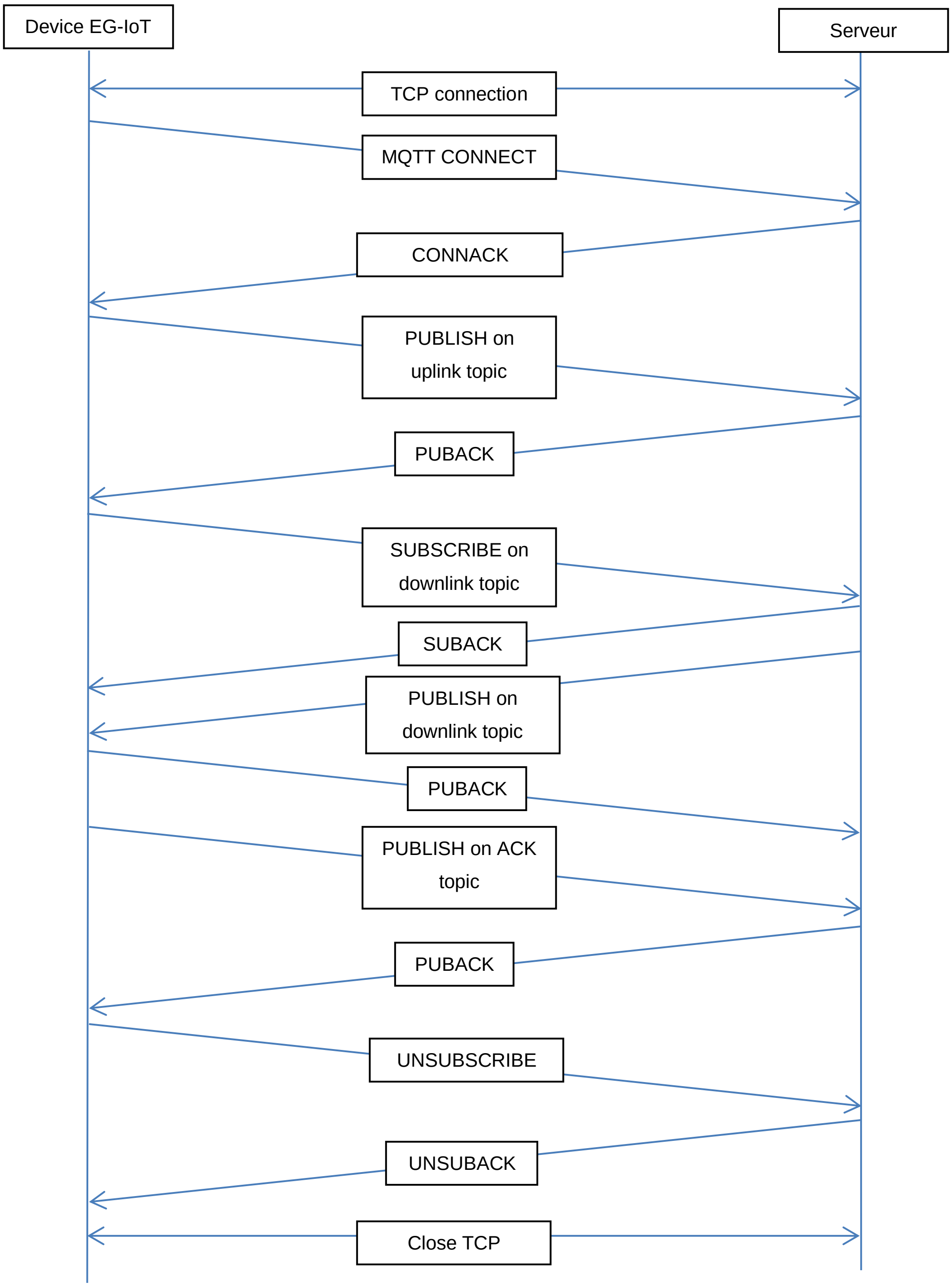
5.5.4 MQTT remote control

To enable MQTT downlink, the parameter "Topic request" must be set and the downlink timeout needs to be a non-null value.
To receive an acknowledgement, the parameter "Topic ACK", must be set.

- ⇨ MQTT topic to subscribe: receive configuration update requests.
- ⇨ MQTT topic to publish: respond to a configuration update request. If publish config is empty no acknowledgement, otherwise, the JSON structure received on the subscribed topic is resent with updated value of the JSON parameter.

The MQTT data must be JSON.

➢ **Exchange between server and device:**



5.5.5 TCP/UDP remote control

To enable TCP/UDP downlink, the parameter "Remote control" must be set and the TCP timeout needs to be a non-null value.

Two data formats must be used, specific structure in text format or JSON format, both using the Hayes special register mechanism described in previous chapter.

5.5.5.1 TCP/UDP downlink using specific structure in text format

The format used is the same as the Hayes special register mechanism. '@' char is before SX.Y and the command needs to terminate with CR and/or LF character. The read command may be used.

- Write command :

Command: `"@S1.5=00:15:00\r"`

Response: `"@S1.5:OK\r\n", or "@S1.5:KO\r\n"`

- Read command:

Command: `"@S1?\r"`

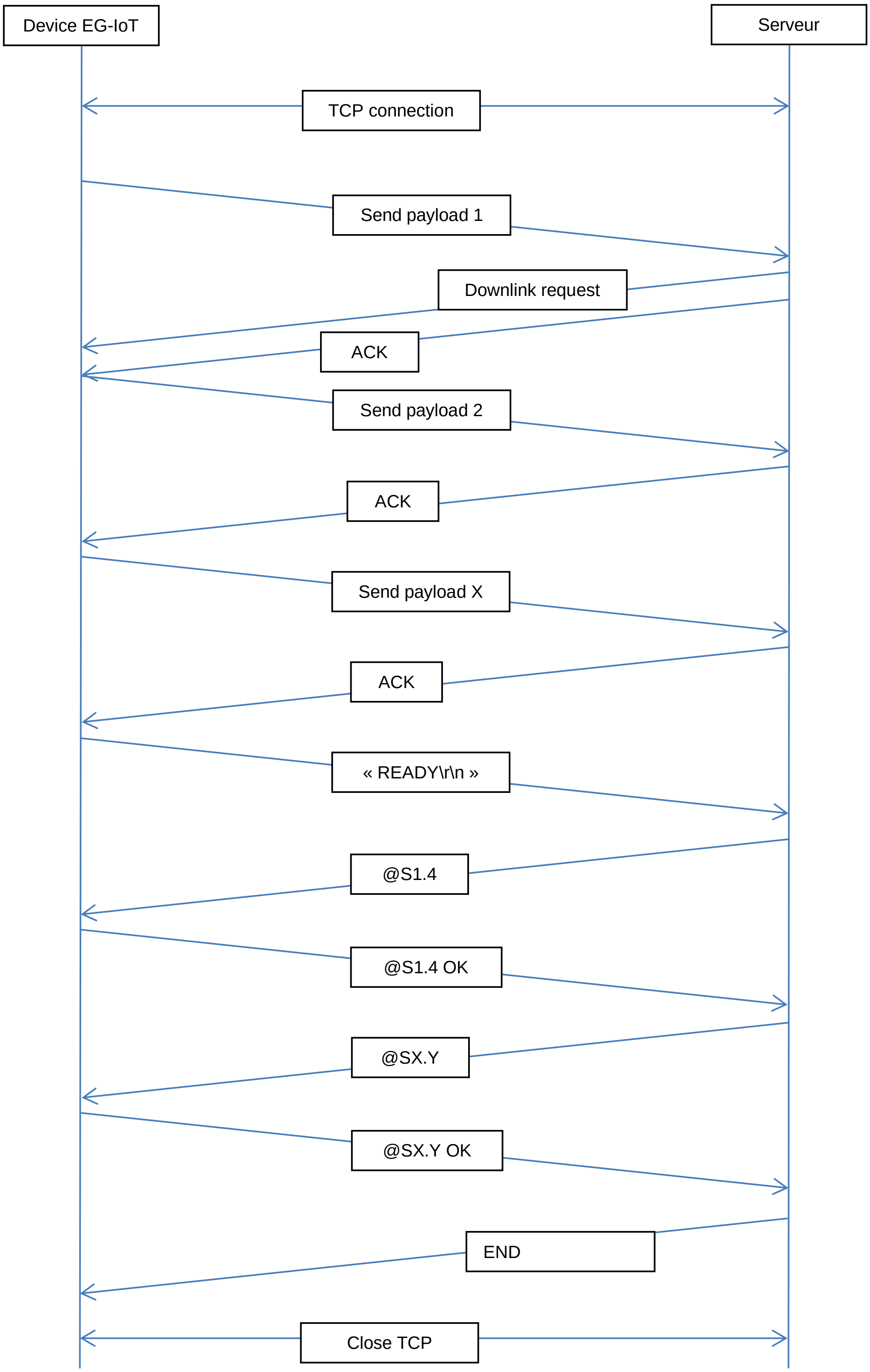
Response: `"@S1: 0;11/09/2019;10:56;99999;I,6;00:10:00;0;1;98;0;3931;27;4E81;V2.11.4b1"`

5.5.5.2 TCP/UDP downlink using JSON format

The format used is the same as described in previous chapter "JSON format for downlink".
More than one JSON structure can be received consecutively and an acknowledge JSON data structure is sent.
CR and LF characters are added after each JSON data structure.

5.5.5.3 Exchange between server and device

- ⇨ Server send downlink request string at the device connexion
- ⇨ To be sure of downlink request reception by device, send downlink request instead of TCP/UDP acknowledge or before ACK.
- ⇨ When the device is ready to receive downlink command, the device send "READY\r\n"
- ⇨ The server sends the downlink command as described before, in text or JSON.
- ⇨ At the end of downlink, the server sends "END\r\n" or closes the socket.



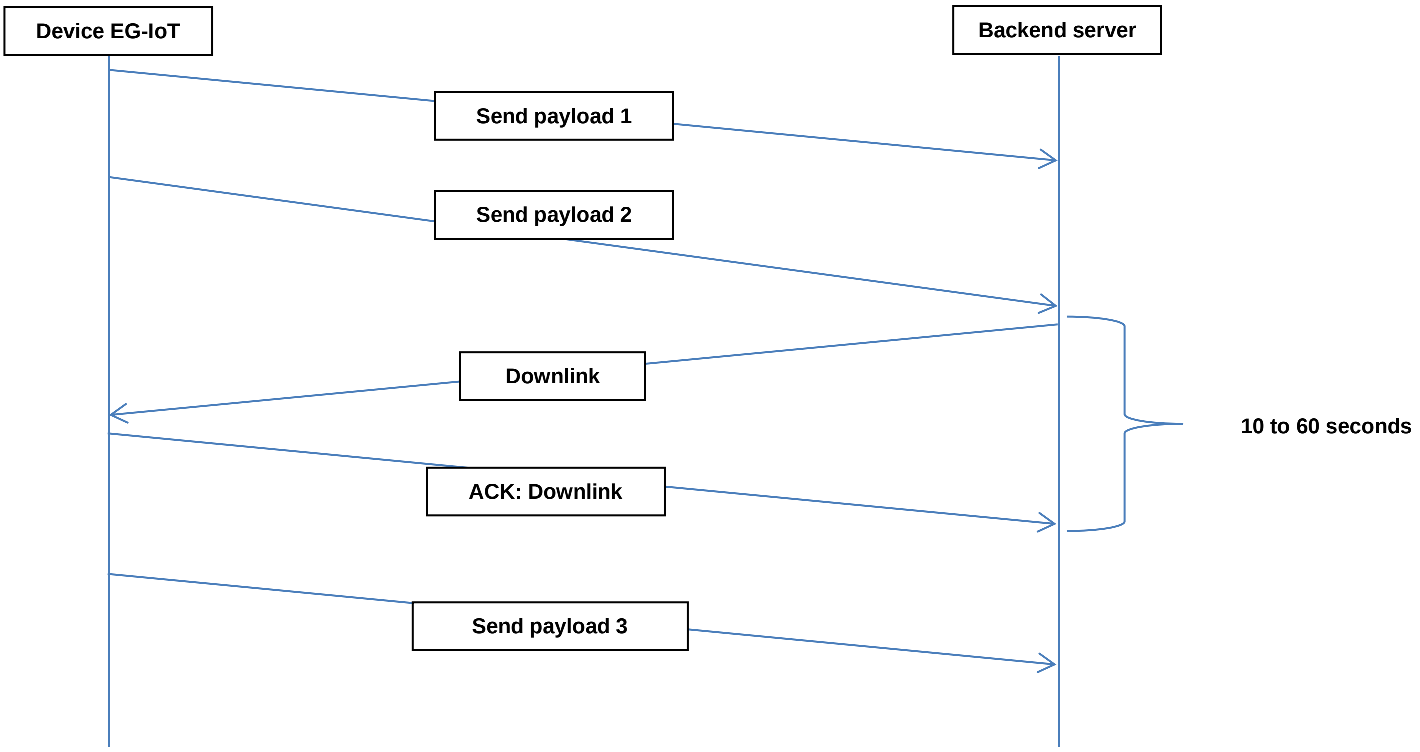
5.6 LPWAN downlink

5.6.1 LoRa Downlink

5.6.1.1 Description

The downlink data will be received at each uplink if present.

Each time the EG-IoT receives a downlink, it sends it back in less than 60 seconds to acknowledge it.



5.6.1.2 Example

In this example, the EG-IoT is in remote reading with a keep alive configured to 60 minutes.

The aim is to drive the output.

Time	Uplink payload	Downlink payloa	Comments
12:00:00	FE000000005E4E87629F4000		Init Frame, Output state : 0
13:00:00	11000000005ADE58629E4000		Keep-Alive, Output state : 0
14:00:00	11000000005ADE58629E4000		Keep-Alive, Output state : 0
14:00:04		40C0000000000000	Downlink send by user to set the ouput
14:00:30	40C0000000000000		ACK : a downlink has been received by the EG-IoT
15:00:30	11000000005E2E6D629F4040		Keep-Alive, Output state : 1
15:00:35	FF		Dummy uplink
16:00:35	11000000005E2E6D629F4040		Keep-Alive, Output state : 1
17:00:35	11000000005E2E6D629F4040		Keep-Alive, Output state : 1
17:00:39		4080000000000000	Downlink send by user to reset the ouput
17:01:00	4080000000000000		ACK : a downlink has been received by the EG-IoT
18:01:00	11000000005ADE58629E4000		Keep-Alive, Output state : 0
19:01:00	11000000005ADE58629E4000		Keep-Alive, Output state : 0

5.6.2 Sigfox Downlink

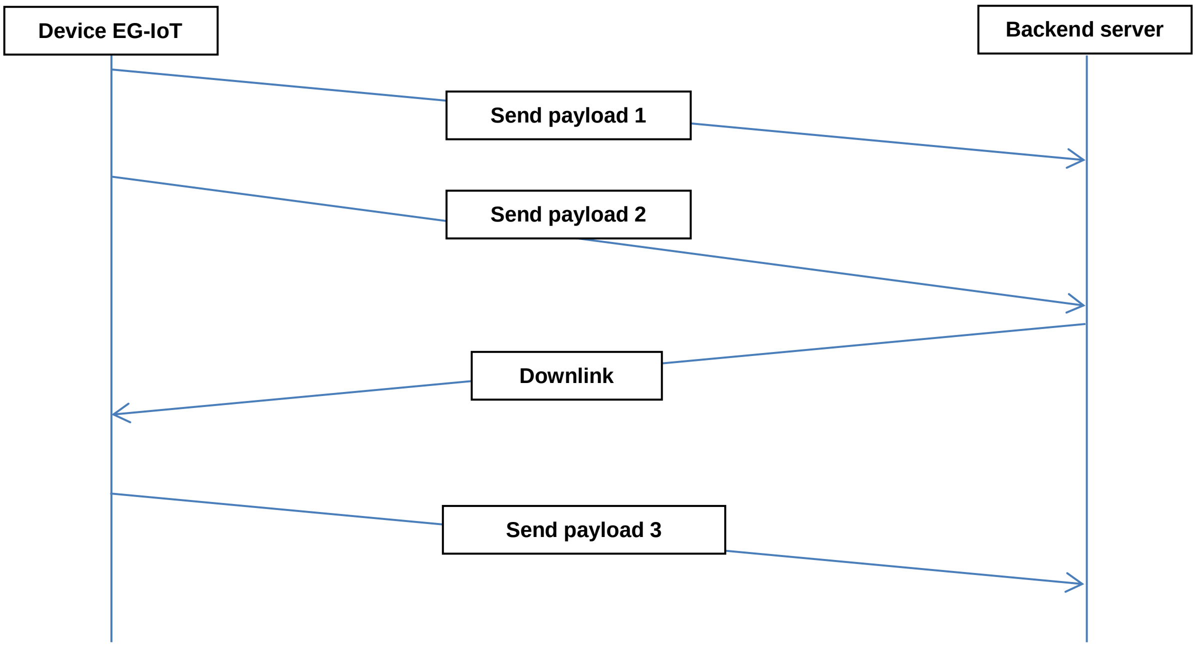
5.6.2.1 Description

sWith Sigfox protocol, EG-IoT asks for a downlink message to the network.

Ask frequency is defined in tab LPWAN, field “Download interval (m)” (Default = 0 = Downlink disable)

Currently Sigfox allows 144 uplinks and 4 downlinks per day.

For these reasons, the EG-IoT sends applications payloads rather than acknowledging downlinks.



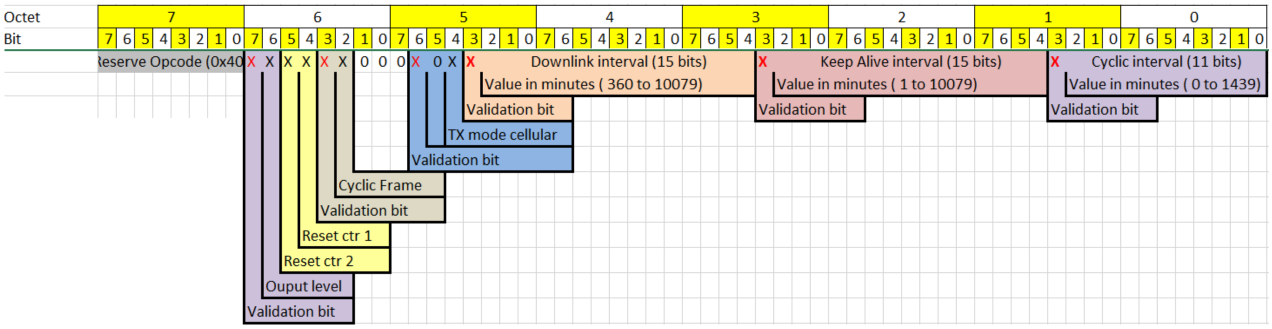
5.6.2.2 Example

In this example, the EG-IoT is in remote reading with a keep alive configured to 60 minutes and Sigfox downlink interval is set to 360 minutes.
The aim is to drive the output.

Time	Uplink payload	Downlink payloa	Comments
12:00:00	FE000000005E4E87629F4040		Init Frame, Output state : 1, DOWNLINK ASK BY EG-IoT
12:00:30		4080000000000000	Downlink send to reset the ouput
13:00:30	11000000005ADE58629E4000		Keep-Alive, Output state : 0
14:00:30	11000000005ADE58629E4000		Keep-Alive, Output state : 0
15:00:30	11000000005ADE58629E4000		Keep-Alive, Output state : 0
16:00:30	11000000005ADE58629E4000		Keep-Alive, Output state : 0
17:00:30	11000000005ADE58629E4000		Keep-Alive, Output state : 0
18:00:30	11000000005ADE58629E4000		Keep-Alive, Output state : 0, DOWNLINK ASK BY EG-IoT
18:01:00		40C0000000000000	Downlink send by user to set the ouput
19:00:00	11000000005E2E6D629F4040		Keep-Alive, Output state : 1

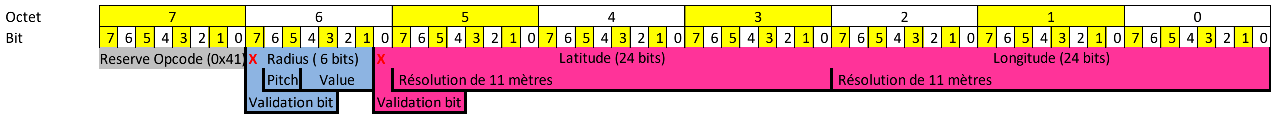
5.6.3 LPWAN Downlink Payload

⇨ Downlink with opcode 0x40 : general configuration



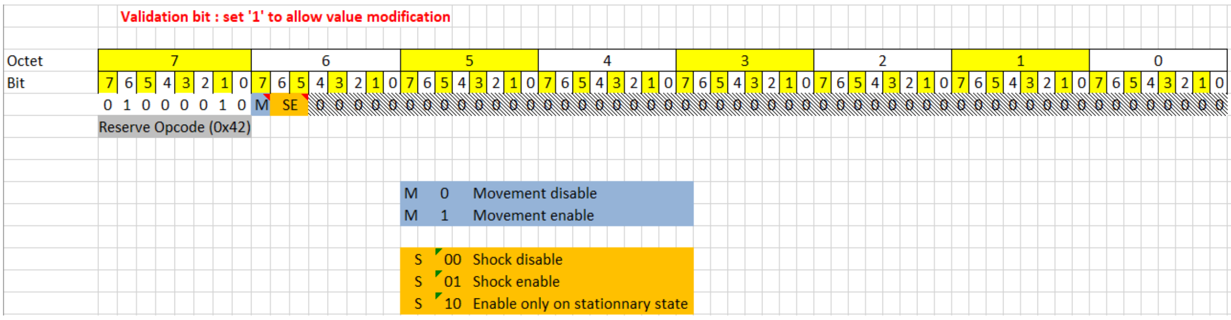
⇨ Downlink with opcode 0x41: geofencing

Validation bit : set '1' to allow value modification

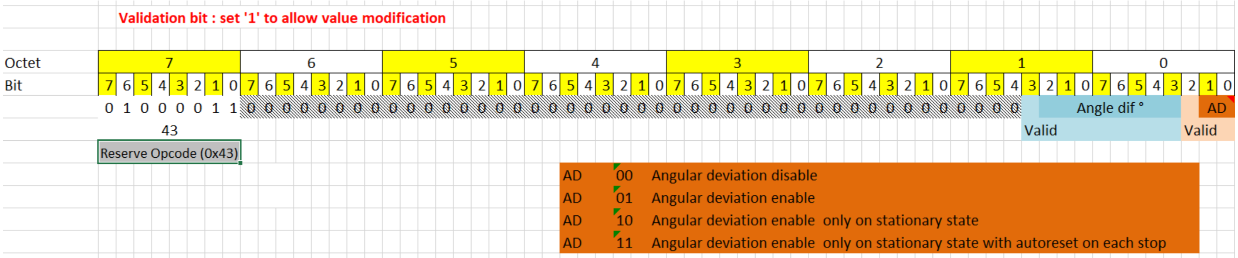


Value	Pitch	Min-Max
0 0	=> 50 m	100-900
0 1	=> 100 m	100-1500
1 0	=> 200 m	200-3000
1 1	=> 400 m	400-6000

⇨ Downlink with opcode 0x42: Accelerometer (for thresholds please contact **Ercogener**)

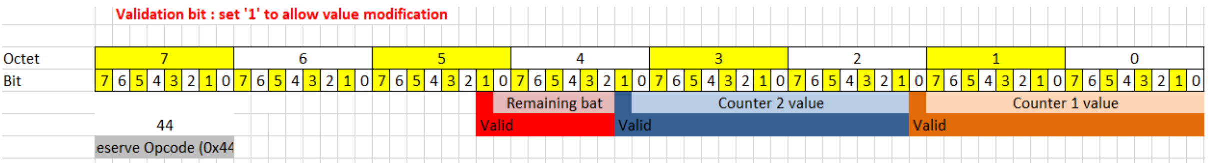


⇨ Downlink with opcode 0x43: Angle

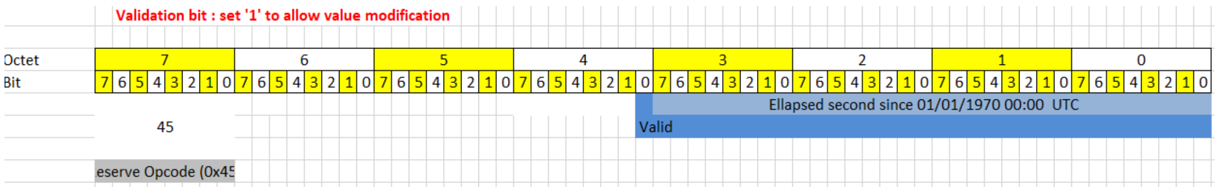


To store a new angular reference before next transmission, send: 0x4300001FFFFFF000

⇨ Downlink with opcode 0x44: Counter value & % remaining battery



⇨ Downlink with opcode 0x45: Date & Time



5.7 Update of embedded software with serial cable

See EG_1152_Ease_EG_IoT_Config_V2.15.2_UG_XXX_UK chapter "Update"

It is strongly recommended to perform "Erase Flash objects"

WARNING, in order to avoid any blockage involving a factory reset of your device, please follow the following precautions:

- the serial cable should not be disconnected,
- the power supply should not be disconnected,
- Ease_EG-IoT_Config application should not be closed,
- do not move the product.

After an update check the product configuration!

ANNEX 1 - MQTT protocol and JSON structure

Specific parameters of the MQTT protocol

MQTT server	MQTT server address to connect. IP address or URL
MQTT port	MQTT connection port
MQTT login	Login for the MQTT connection
MQTT password	Password for the MQTT connection
MQTT topic	MQTT topic to use in data publishing
MQTT client id	MQTT client id used in MQTT connection

For access to the MQTT server, consult your provider.

For the Live Objects server from Orange:

https://liveobjects.orange-business.com/doc/html/lo_manual.html#MQTT_MODE_DEVICE

The format and content of the data published by the MQTT is configured in the JSON string field:

- JSON data to publish in MQTT protocol. The format is detailed below.

MQTT

MQTT protocol version: 3.1.1

QOS: 1

Frames:

- CONNECT
- CONNACK
- PUBLISH
- PUBACK
- DISCONNECT
- SUBSCRIBE
- SUBACK

Configuration parameter:

- topic
- user
- password
- client_id

JSON data configuration

The JSON data may be sent either in TCP socket or published by MQTT. The JSON format is not detailed here but the use and data insertion information may be found here: **https://en.wikipedia.org/wiki/JSON**

To resume, the JSON format is an object representation, containing:

- A collection of name / value couples
- A list of ordered values

The values may be:

- String
- Boolean
- Integer
- Float
- Null value
- objects

The JSON structure is dynamic; in this case, the user must build and configure his or her own JSON structure. A default JSON structure is stored in non-volatile memory and reload on restore default configuration.

The user is free to define one's own JSON structure according to required use case. The name of the objects and content is free of choice. The application of the EG-IoT box provides a list of dynamic variables that the user can insert into the values of the objects. Dynamic variables are represented by a name, starting and ending with the character '$'. The variable is case-sensitive.

A JSON value can contain a value other than a dynamic variable but can also contain a concatenation of both.

During construction, the dynamic variables will be inserted in the JSON structure according to their predefined type or as a string if the dynamic variable is inserted in a text or if it is itself a text.

For example, "$TEMP$ ", will be converted to "22 " and will be of string type due to the character space ' ' present in the value of the original variable.

There are however the following limitations:

- The JSON structure accepts displayable characters from the 7-bit ASCII table; the extended ASCII table has compatibility issues.
- The dynamic fields of the structure can only be present in the value of the JSON objects and will be replaced by the value of the measure at the time of sending.
- The names of objects in the JSON structure are not dynamic; it is not possible to insert an **Ercogener** variable.

During configuration, the structure may be indented and contain the end-of-line characters CR and LF. However, for reasons of storage size and processing time, these are deleted when the structure is saved.

When configuring the JSON structure, a check is made and a sample JSON frame is displayed, containing test values.

Example of JSON structure for a tracking use case, default value:

```
{
        "s": "urn:lo:nsid:1114_TEST_BE:$IMEI$",
        "ts": "$DATE_TIME$",
        "m": "init_model",
        "loc": [ "$LAT$", "$LON$" ],
        "v":
        {
                "temperature":
                {
                        "value": "$TEMP$",
                        "unit": "degC"
                },
                "battery":
                {
                        "value": "$VBAT_V$",
                        "unit": "V",
                "remaining": "$BAT_CAP$"
                },
                "GPS":
                {
                        "validPosition": "$GNSS_FIX_INT$",
                        "speed":
                        {
                        "value": "$SPEED$",
                        "unit": "kmh"
                        },
                        "onMove": "$OPCODE_INT$"
                },
                "network":
                {
                        "rsrp": "$SIG_STRENGTH$",
                        "rsrq": "$SIG_QUAL$"
                }
        },
        "t": [ "Ercogener", "Saumur" ]
}
```

Example of MQTT published data with default value, for a tracking use case:


```
{
	"s": "urn:lo:nsid:1114_TEST_BE:123456789011121",
	"ts": "2018-01-01T12:01:20.0Z",
	"m": "init_model",
	"loc": [
		47.2113,
		-0.06054
	],
	"v": {
		"temperature": {
			"value": 26,
			"unit": "degC"
		},
		"battery": {
			"value": 3.998,
			"unit": "V",
			"remaining": 0
		},
		"GPS": {
			"validPosition": 1,
			"speed": {
				"value": 0.0,
				"unit": "kmh"
			},
			"onMove": 17
		},
		"network": {
			"rsrp": 10,
			"rsrq": 10
		}
	},
	"t": [
		"Ercogener",
		"Saumur"
	]
}
```

Example of JSON structure for a metering use case:


```
{
	"s": "urn:lo:nsid:1114_TEST_BE:$IMEI$",
	"ts": "$DATE_TIME$",
	"m": "init_model",
	"v":
	{
		"opcode": "$OPCODE_INT$",
		"temperature":
		{
			"value": "$TEMP$",
			"unit": "degC",
			"alarm": "$ALARM_INT$"
		},
		"battery":
		{
			"value": "$VBAT_V$",
			"unit": "V",
			"remaining": "$BAT_CAP$"
		},
		"input1":
		{
			"state": "$IN1_STATE$",
			"counter": "$CPT1_VALUE$"
		},
		"input2":
		{
			"state": "$IN2_STATE$",
			"counter": "$CPT2_VALUE$"
		},
		"network":
		{
			"rsrp": "$SIG_STRENGTH$",
			"rsrq": "$SIG_QUAL$"
		}
	},
	"t": [
		"Ercogener",
		"Saumur"
	]
}
```

Example of JSON structure for an angular deviation use case:


```
{
	"s": "urn:lo:nsid:1114_TEST_BE:$IMEI$",
	"ts": "$DATE_TIME$",
	"m": "init_model",
	"loc": [ "$LAT$", "$LON$" ],
	"v":
	{
		"temperature":
		{
			"value": "$TEMP$",
			"unit": "degC"
		},
		"battery":
		{
			"value": "$VBAT_V$",
			"unit": "V",
			"remaining": "$BAT_CAP$"
		},
		"GPS":
		{
			"validPosition": "$GNSS_FIX_INT$",
			"speed":
			{
				"value": "$SPEED$",
				"unit": "kmh"
			}
		},
		"AXE":
		{
			"X":
			{
				"value": "$ANGLE_X$",
				"unit": "deg"
			},
			"Y":
			{
				"value": "$ANGLE_Y$",
				"unit": "deg"
			},
			"Z":
			{
				"value": "$ANGLE_Z$",
				"unit": "deg"
			}
		},
		"onMove": "$OPCODE_INT$"
	},
	"network":
	{
		"rsrp": "$SIG_STRENGTH$",
		"rsrq": "$SIG_QUAL$"
	},
	"t": [ "Ercogener", "Saumur" ]
}
```

Example of JSON structure for a 2G/3G cell-id use case:


```
{
	"s": "urn:lo:nsid:1114_TEST_BE:$IMEI$",
	"ts": "$DATE_TIME$",
	"network":
	{
		"rsrp": "$SIG_STRENGTH$",
		"rsrq": "$SIG_QUAL$"
	},
	"cellid":
	[
		{
			"cell_0": {
				"mcc": "$MCC0$",
				"mnc": "$MNC0$",
				"rat": "$RAT0$",
				"lac": "$LAC0$",
				"cellID": "$CID0$",
				"rssi": "$RSSI0$"
			}
		},
		{
			"cell_1": {
				"mcc": "$MCC1$",
				"mnc": "$MNC1$",
				"rat": "$RAT1$",
				"lac": "$LAC1$",
				"cellID": "$CID1$",
				"rssi": "$RSSI1$"
			}
		},
		{
			"cell_2": {
				"mcc": "$MCC2$",
				"mnc": "$MNC2$",
				"rat": "$RAT2$",
				"lac": "$LAC2$",
				"cellID": "$CID2$",
				"rssi": "$RSSI2$"
			}
		},
		{
			"cell_3": {
				"mcc": "$MCC3$",
				"mnc": "$MNC3$",
				"rat": "$RAT3$",
				"lac": "$LAC3$",
				"cellID": "$CID3$",
				"rssi": "$RSSI3$"
			}
		},
		{
			"cell_4": {
				"mcc": "$MCC4$",
				"mnc": "$MNC4$",
				"rat": "$RAT4$",
				"lac": "$LAC4$",
				"cellID": "$CID4$",
				"rssi": "$RSSI4$"
			}
		}
	],
	"t": [
		"Ercogener",
		"Saumur"
	]
```

Example of JSON structure for a LTE-M1 cell-id use case:


```
{
	"s": "urn:lo:nsid:1114_TEST_BE:$IMEI$",
	"ts": "$DATE_TIME$",
	"m": "init_model",
	"loc": [ "$LAT$", "$LON$" ],
	"v":
	{
		"temperature":
		{
			"value": "$TEMP$",
			"unit": "degC"
		},
		"battery":
		{
			"value": "$VBAT_V$",
			"unit": "V",
			"remaining": "$BAT_CAP$"
		},
		"GPS":
		{
			"validPosition": "$GNSS_FIX_INT$",
			"speed":
			{
				"value": "$SPEED$",
				"unit": "kmh"
			},
			"onMove": "$OPCODE_INT$"
		},
		"network":
		{
			"rsrp": "$SIG_STRENGTH$",
			"rsrq": "$SIG_QUAL$"
		}
	},
	"cellid":
	[
		{
			"cell_0":
			{
				"mcc": "$MCC0$",
				"mnc": "$MNC0$",
				"rat": "$RAT0$",
				"lac": "$LAC0$",
				"cellID": "$CID0$",
				"rssi": "$RSSI0$"
			}
		}
	],
	"t": [ "Ercogener", "Saumur" ]
}
```

Dynamic field list:

Variable	Function	Type	Values
$GNSS_FIX_INT$	GNSS Status	Integer	0 : invalid 1 : GPS Fix (2D/3D) 2 : DGPS Fix
$GNSS_FIX_STRING$	GNSS Status	string	"Invalid" " GPS Fix (2D/3D)" " DGPS Fix "
LAT	Latitude of the position, given by GPS	double	In degrees
LON	Longitude of the position, given by GPS	double	In degrees
ALT	Altitude of the position, given by GPS	double	In meters
$ACCURACY_MODE$	Accuracy working mode	string	"DOP" (HDOP accuracy) "Precision" (Meter accuracy)
$ACCURACY_STRING$	GPS accuracy	string	"Very good" (HDOP <= 2 Excellent / Accuracy <=20m) "Good" (2 < HDOP <= 5 correct / 20m< Accuracy <=50m) "Moderate" (5 < HDOP <= 9 moyen / 50m< Accuracy <=100m) "Poor" (9 < HDOP très faible / 100m< Accuracy)
$ACCURACY_DEC$	GPS accuracy	double	Value (depends on the configuration of the precision management mode) : • Real HDOP value, from 0 to 99, • Accuracy value in meters.
$ACCURACY_INT$	GPS accuracy	Integer	Interpreted value of HDOP : • 0 (HDOP <= 2 Excellent / Precision<=20m) • 1 (2 < HDOP <= 5 correct / 20m<Precision<=50m) • 2 (5 < HDOP <= 9 medium / 50m<Precision<=100m) • 3 (9 < HDOP very low / 100m<Precision)
$SPEED$	Speed of the device	Double	In km/h
$TEMP$	Temperature inside of the device in °C	Integer	-40 to +85°C
$VBAT_V$	Batterie voltage	Double	In V
$VBAT_MV$	Batterie voltage	Integer	In mV
BAT_CAP	Remaining battery capacity	Integer	In %
$TIMESTAMP$	Current UTC date and time of the device	Integer	Format : Linux timestamp (seconds since 01/01/1970 00:00:00 UTC)
$DATE_1$	Current UTC date and time of the device	String	Format : dd/mm/yyyy
$DATE_2$	Current UTC date and time of the device	String	Format : mm/dd/yyyy
$DATE_3$	Current UTC date and time of the device	String	Format : dd.mm.yyyy
$DATE_4$	Current UTC date and time of the device	String	Format : yy/mm/dd
$DATE_TIME$	Current UTC date and time of the device	String	Format : yyyy-mm-ddThh:mm:ss.0Z
$TIME$	Current UTC date and time of the device	String	Format : hh:mm:ss
$OPCODE_INT$	Frame type	Integer	Values for tracking mode ➢ 0 : initialisation ➢ 16 : cyclic ➢ 17 : keep alive ➢ 128 : movement stop ➢ 132 : movement start ➢ 138 : shock ➢ 160 : exit zone 0 ➢ 192 : entry zone 0 ➢ 255 : Dummy Values for telemetering mode ➢ 16 : cyclic ➢ 17 : keep alive ➢ 23 : Event or alarm ➢ 254 : initialisation ➢ 255 : Dummy
$OPCODE_STRING$	Frame type	String	Values for tracking mode ➢ "Initialisation" (0) ➢ "Cyclic" (16) ➢ "Keep alive" (17) ➢ "Movement stop" (128) ➢ "Movement start" (132) ➢ "Shock" (138) ➢ "Exit zone 0" (160) ➢ "Entry zone 0" (192) ➢ "Dummy" (255) Values for telemetering mode ➢ "Cyclic " (16) ➢ "Keep alive " (17) ➢ "Event" (23) ➢ "Initialisation" (254) ➢ "Dummy" (255)
$IMEI$	GSM IMEI	String	
DEV_ID	Device identifier	String	Default value is the IMEI value.

Variable	Function	Type	Values
$LPWAN_ID$	LPWAN Module Identifier	String	
$HW_CAPACITY$	Hardware capacity of the EG-IoT device	String	
$SIG_STRENGTH$	Signal strength, RSSI (2G/3G) ou RSRP (LTE)	Integer	See ANNEX 2 - Other information: GSM (RSSI/QUAL) LTE (RSRQ/RSRP)
SIG_QUAL	Signal quality, QUAL (2G/3G) ou RSRQ (LTE)	Integer	See ANNEX 2 - Other information: GSM (RSSI/QUAL) LTE (RSRQ/RSRP)
$URAT$	Radio access technology used for the current cellular connection	Integer	0: GSM 1: GSM/UMTS 2: UMTS 3: LTE 4: GSM/UMTS/LTE 5 GSM/LTE 6: UMTS/LTE 7: LTE CAT M1 8: LTE CAT NB1 9: GPRS/eGPRS
$IN1_STATE$	Digital input 1 state	Integer	0 or 1
$IN2_STATE$	Digital input 2 state	Integer	0 or 1
$CPT1_VALUE$	Counter value of the digital input 1	Integer	
$CPT2_VALUE$	Counter value of the digital input 2	Integer	
$ALARM_HEX$	Hexadecimal value of the alarms and device state	String	Bits mask : **b0** Status Input 1 **b1** Status Input 2 **b2** Alarm counter 1 0 < Threshold 1 ≥ Threshold **b3** Alarm counter 2 0 < Threshold 1 ≥ Threshold **b4** Alarm lox-Temp 0 < Threshold 1 ≥ Threshold **b5** Alarm high-Temp 0 < Threshold 1 ≥ Threshold **b6, b7** Reserved Example : "19" (0x19): low temperature alarm – counter 2 alarm – input 2 = 0 – input 1 = 1
$ALARM_INT$	Hexadecimal value of the alarms and device state	Integer	Bits mask : **b0** Status Input 1 **b1** Status Input 2 **b2** Alarm counter 1 0 < Threshold 1 ≥ Threshold **b3** Alarm counter 2 0 < Threshold 1 ≥ Threshold **b4** Alarm lox-Temp 0 < Threshold 1 ≥ Threshold **b5** Alarm high-Temp 0 < Threshold 1 ≥ Threshold **b6, b7** Reserved Example : "25" 25: low temperature alarm – counter 2 alarm – input 2 = 0 – input 1 = 1
$CIDX$	X index of cell-id from 0 to 4	Integer	Cell identifier index X
$LACX$	X index of cell-id from 0 to 4	Integer	Location Area Code of cell-id X
$MCCX$	X index of cell-id from 0 to 4	Integer	Mobile Country Code of cell-id X
$MNCX$	X index of cell-id from 0 to 4	Integer	Mobile Network Code of cell-id X
$RATX$	X index of cell-id from 0 to 4	String	Radio Access Technology of cell-id X, value: “2G” “3G” “4G”
$RSSIX$	X index of cell-id from 0 to 4	Integer	Signal strength of cell-id X The measured signal level (RSSI) shall be mapped to an RXLEV value between 0 and 63, as follows: ➢ RXLEV0=less than-110 dBm. ➢ RXLEV1=-110 dBmto-109 dBm. ➢ RXLEV2=-109 dBmto-108 dBm. ➢ […] ➢ RXLEV62=-49 dBmto-48 dBm. ➢ RXLEV63=greater than-48 dBm. ➢ 255 not measured
OUT	Digital output state value	Integer	0 or 1

ANNEX 2 - Other information: GSM (RSSI/QUAL) LTE (RSRQ/RSRP)

GSM RSSI (Received Signal Strength Indication)	
0	Less than -110.5 dBm
1..30	From -110.5 dBm to -51 dBm
31	More than -51 dBm
99	Unknow

	GSM QUALity		
	2G – CSD data / GPRS **Bit Error Rate**	**2G - EGPRS** **Bit Error Probability**	**3G - UMTS** **Energy per Chip/Noise**
0	BER < 0.2%	28 <= BEP <= 31	ECNo >= 44
1	0.2% < BER < 0.4%	24 <= BEP <= 27	38 <= ECNo < 44
2	0.4% < BER < 0.8%	20 <= BEP <= 23	32 <= ECNo < 38
3	0.8% < BER < 1.6%	16 <= BEP <= 19	26 <= ECNo < 32
4	1.6% < BER < 3.2%	12 <= BEP <= 15	20 <= ECNo < 26
5	3.2% < BER < 6.4%	8 <= BEP <= 11	14 <= ECNo < 20
6	6.4% < BER < 12.8%	4 <= BEP <= 7	8 <= ECNo < 14
7	BER > 12.8%	0 <= BEP <= 3	ECNo < 8
99		unknown	

LTE - RSRQ (Reference Signal Received Quality)	
0	Less than -19 dB
1..33	From -19.5 dB to -3.5 dB step 0.5 dB
34	More than -3 dB
255	Unknown

LTE - RSRP (Reference Signal Received Power)	
0	Less than -141 dBm
1..96	From -140 dBm to -45 dBm step 1 dBm
97	More than -44 dBm
255	Unknown

ANNEX 3 - Certificates and private key configuration

Configuration

The X.509 certificates and private key may be in PEM format.

In the menu, to manage the certificate, 4 choices:

- Send the certificate or key in PEM format, without any other characters and then send "ctrl-z" (0x1A) character to terminate.
- ‘D’: disable the use of the certificate or key
- ‘E’: enable the use of the certificate or key
- ‘R’: remove the certificate or key

The certificate or key will not be readable from the device but the md5 will be displayed.

Private Key

PKCS1/PKSC8 RSA key formats are supported with the following restriction:

- LTE-CAT1 modules:
  - PKCS1 RSA encrypted key not supported
- Other modules:
  - Private key password not supported
  - PKCS1 RSA encrypted key not supported
  - PKCS8 unencrypted and encrypted key not supported

Example of how to generate private key with password with Linux commands:

```
[user@localhost private_key]$ ssh-keygen -t rsa -f test_rsa
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in test_rsa.
Your public key has been saved in test_rsa.pub.
The key fingerprint is:
SHA256:nn/2XDXdFSUltbsJsz7op2Hi/oE/ycIpsKOfQs9ofyI user@localhost.localdomain
The key's randomart image is:
+---[RSA 2048]----+
|            o+=|
|             oo|
|             ..|
|              =|
|       S    o o+|
|  . . . ..   + =|
| . + o +oo+o. o.|
|  E *.o.=+=*+ . |
| ..*++ oo=B+o+  |
+----[SHA256]-----+
[user@localhost private_key]$ ls -l
-rw------- 1 user user 1679 11 oct.  12:11 test_rsa
-rw-r--r-- 1 user user  408 11 oct.  12:11 test_rsa.pub
[user@localhost private_key]$ openssl pkcs8 -topk8  -v2 des3 -in test_rsa -out test_rsa_pkcs8
Enter Encryption Password:
Verifying - Enter Encryption Password:
```